TAMPEREEN TEKNILLINEN YLIOPISTO

JANNE LIUTTU

ARVOSTUSALGORITMIT VERKOSTOANALYYSISSA

Diplomityö

Tarkastajat: Prof. Seppo Pohjolainen (TTY) ja tutkija Jukka Huhtamäki (TTY)
Tarkastaja ja aihe hyväksytty
Luonnontieteiden ja ympäristötekniikan tiedekuntaneuvoston kokouksessa
15. elokuuta 2012

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Teknis-luonnontieteellinen koulutusohjelma
JANNE LIUTTU: Arvostusalgoritmit verkostoanalyysissa
Diplomityö, 84 sivua
Elokuu 2012
Pääaine: Matematiikka
Tarkastajat: Prof. Seppo Pohjolainen (TTY) ja tutkija Jukka Huhtamäki (TTY)
Avainsanat: Verkostoanalyysi, graafiteoria, matriisilaskenta, Pagerank, HITS

Internetin räjähdysmäinen kasvu on tehnyt entistä suuremmaksi haasteeksi löytää merkitykselliset ja luotettavat sivustot harmaasta massasta. Tähän tarkoitukseen on kehitetty erilaisia arvostusalgoritmeja, joilla pyritään asettamaan eri sivustot objektiiviseen paremmuusjärjestykseen. Merkittävämmät tällaiset algoritmit ovat Pagerank ja HITS, jotka molemmat ovat lähtöisin internetin hakukoneista, ja perustuvat internetin linkkirakenteeseen. Näiden algoritmien hyödyntämismahdollisuudet eivät kuitenkaan rajoitu ainoastaan internetin hakukoneisiin, ja näillä on käytännössä mahdollista tarkastella lähes minkälaista verkostoa tahansa.

Näiden arvostusalgoritmien matemaattinen käsittely perustuu pitkälti graafiteoriaan sekä matriisilaskentaan, graafiteorian tarjoten työkalut verkostojen mallinnukseen ja visualisointiin, ja matriisilaskennan luodessa pohjan arvostusten laskemiselle. Nämä kaksi matematiikan osa-aluetta nivoutuvat siististi yhteen, muodostaen elegantin kokonaisuuden arvostusalgoritmien käsittelylle.

Tässä diplomityössä perehdytään eri arvostusalgoritmeihin, sekä näiden matemaattiseen taustaan. Esimerkkidatana käytetään Tampereen teknillisen yliopiston vuoden 2010 opinto-oppaan kurssien esitietoketjuja, joista muodostuu käsiteltävä verkosto. Ilmiönä esitietoketjut eroavat jonkin verran internetin linkkirakenteesta, mutta tuloksissa havaitaan että algoritmit toimivat hyvin myös tämänkaltaisen verkoston tapauksessa. Pääsääntöisesti eri algoritmit tuottavat saman suuntaisia arvostuksia eri kursseille, mutta eri algoritmit painottavat kukin hieman eri asioita. Näin ollen eri algoritmien toiminnan tunteminen on ensiarvoisen tärkeää, kun pohditaan näiden hyödyntämistä eri ilmiöiden tarkastelussa.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Science and Engineering
JANNE LIUTTU : Ranking algorithms in network analysis
Master of Science Thesis, 84 pages
August 2012
Major: Mathematics
Examiner: Prof. Seppo Pohjolainen (TUT) and researcher Jukka Huhtamäki (TUT)
Keywords: Network analysis, graph theory, matrix algebra, Pagerank, HITS

The fast expansion of the internet has made it a bigger challenge to find the most relevant and reliable pages from the vastness of pages. A bunch of different ranking algorithms have been developed to solve this problem, and their goal is to objectively rank the pages. The two most significant algorithms are the Pagerank and the HITS, which both originate from the search engines of the web, and use the link structure of the web as a base of their calculation. The utilization of these algorithms is however not limited to the internet, and they can be used to examine almost any kind of a network.

The mathematical basis of these algorithms is mainly graph theory and matrix algebra. Graph theory provides framework for the modeling and visualization of the networks, and matrix algebra can be used to calculate the rankings. These two branches of mathematics blend in neatly, forming an elegant environment for the handling of the ranking methods.

This Master's Thesis describes different ranking algorithms, and the mathematics behind them. As a case study we examine the prerequisites of different courses in the 2010 course catalog from Tampere university of technology. These prerequisites form a network, which differs slightly from the link structure of the internet, but we can see that the algorithms work well also in this case. The different algorithms produce mainly pretty similar rankings between different courses, but each of them emphasize slightly different things. Therefore knowing the behavior of different algorithms is essential when considering utilizing them when analyzing different phenomena.

ALKUSANAT

Tämä työ on tehty kevään ja kesän 2012 aikana omalla vapaa-ajalla sekä omalla rahoituksella, työskennellessä samanaikaisesti täysipäiväisesti analyytikkona. Kaikki tähän työhön liittyvät tulokset luovutetaan työn valmistumisen yhteydessä Tampereen teknillisen yliopiston matematiikan laitokselle, heidän vapaasti käytettäväkseen ja hyödynnettäväkseen.

Ensimmäiseksi haluan kiittää työni tarkastajia professori Seppo Pohjolaista sekä tutkija Jukka Huhtamäkeä, jotka tarjosivat ohjausta, ajatuksia sekä erilaisia näkökulmia yhtälailla työn teoriaosuuteen, kuin tulosten tulkintaan ja esittämiseen.

Lisäksi haluan kiittää työtovereitani Andumus Oy:ssä, joiden ohjauksessa olen kehittynyt huimasti analyytikkona, löytäen uusia näkökulmia niin päivätyöhöni, kuin myös tämän diplomityön toteuttamiseen.

Lopuksi haluan kiittää perhettäni, sukulaisiani, opiskelutovereitani sekä ystäviäni, joiden kaikkien myötävaikutuksella tästä työstä muovautui lopullisen kaltaisensa. Erityismaininta suotakoon joukkuetovereilleni SBS Soittorasiassa, joiden seurassa sain viettää lukuisia hienoja hetkiä läpi koko opiskeluaikani.

Helsingissä 31. Elokuuta 2012

Janne Liuttu
Strömbergintie 8 A 14
FI-00380 Helsinki
janne.liuttu@andumus.fi

SISÄLLYS

KUVAT

TAULUKOT

TERMIT JA SYMBOLIT

$\mathcal{G}$	Graafi
$\mathcal{G}_d$	Suunnattu graafi
$\mathcal{V}$	Solmujen joukko
$\mathcal{L}$	Kaarien joukko
n_i	Solmu $i = 1 \dots g$
l_i	Kaari $i = 1 \dots m$
(a, b)	Järjestämätön pari
$< a, b >$	Järjestetty pari
$\emptyset$	Tyhjä joukko
$d(n_i)$	Solmun n_i aste
$d_I(n_i)$	Solmun n_i tuloaste
$d_O(n_i)$	Solmun n_i lähtöaste
Δ	Graafin tiheys
$[0, 1]$	Suljettu väli
Q	Graafin modulaarisuus
W	Kulku
W_d	Suunnattu kulku
L	Vieruspistematriisi
l_{ij}	Vieruspistematriisin alkio
L^k	Vieruspistematriisin potenssimatriisi
$[L^k]_{ij}$	Vieruspistematriisin potenssimatriisin alkio
H	Linkkimatriisi
h_{ij}	Linkkimatriisin alkio
I	Identiteettimatriisi
$\sum_{k=1}^{n}$	Summa $K = 1 \dots n$
A^{-1}	Matriisin A käänteismatriisi
A^T	Matriisin A transpoosi
e^A	Matriisin A eksponenttimatriisi
$a!$	Kertoma 1*2*3*...*(a-1)*a
$\|\|x\|\|$	Vektorinormi
$\|a\|$	Itseisarvo
max	Maksimi
λ	Ominaisarvo
x	(Oikeanpuoleinen) ominaisvektori
y^T	Vasemmanpuoleinen ominaisvektori
$p(\lambda)$	Karakteristinen polynomi
$det(A)$	Matriisin A determinantti

$\sigma(A)$	Matriisin A spektri
$\rho(A)$	Matriisin A spektrisäde
$algmult_A(\lambda)$	Ominaisarvon λ algebrallinen kertaluku
$geomult_A(\lambda)$	Ominaisarvon λ geometrinen kertaluku
P	Permutaatiomatriisi
G_i	Spektriprojektiomatriisi
$f()$	Funktio f
$N(A)$	Matriisin A ominaisavaruus
$\prod_{j=1}^{s}$	Tulo $j = 1 \dots s$
$\overline{a}$,a^*	Kompleksikonjugaatti
$n \to \infty$	n lähestyy ääretöntä
$\xrightarrow{n\to\infty}$	Raja-arvo kun n lähestyy ääretöntä
$a \in A$	Alkio a kuuluu joukkoon A
$A \geq 0$	Matriisin A alkiot ovat ei-negatiivisia
$A > 0$	Matriisin A alkiot ovat positiivisia
r	Perronin juuri
min	Minimi
X_t	Satunnaismuuttuja X
S_i	Stokastisen prosessin tila
$P(X\|Y)$	Ehdollinen todennäköisyys
P	Siirtymätodennäköisyysmatriisi
p_{ij}	Siirtymätodennäköisyysmatriisin alkio
e	Vektori, jonka kaikki alkiot ovat ykkösiä
p	Tilajakauma
π	Stationäärinen tilajakauma
$lim_{k\to\infty}$	Raja-arvo kun k lähestyy ääretöntä
O_i	Solmuun n_i osoittavien solmujen joukko
I_i	Niiden solmujen joukko, joihin solmu n_i osoittaa
r^T	Pagerank-arvo
S	Stokastinen linkkimatriisi
a	Nieluvektori
G	Googlematriisi
v	Personointivektori
G_p	Personoitu Googlematriisi
r_p	Personoitu Pagerank
L_c	Käännetty vieruspistematriisi
H_c	Käännetty linkkimatriisi
$h_{c_{ij}}$	Käännetyn linkkimatriisin alkio

b	Lähdevektori
G_c	Käännetty Googlematriisi
c^T	Cheirank-arvo
$\mathcal{N}$	Ympäristögraafi
x	HITS auktoriteettiarvo
y	HITS hubiarvo
L_{row}	Rivinormitettu vieruspistematriisi
L_{col}	Sarakenormitettu vieruspistematriisi
$\mathcal{V}_h$	Hubisolmujen joukko
$\mathcal{V}_a$	Auktoriteettisolmujen joukko
$n_i^{(h)}$	Solmua n_i vastaava solmu hubisolmujen joukossa $\mathcal{V}_h$
$n_j^{(a)}$	Solmua n_j vastaava solmu auktoriteettisolmujen joukossa $\mathcal{V}_a$
$\mathcal{H}$	SALSA-algoritmin graafi
g_H	Solmujen lukumäärä graafissa $\mathcal{H}$
α	Pagerank- ja Cheirank-algoritmien parametri
ξ	HITS-algoritmin parametri
ϵ	Suppenemiskriteeri
log_{10}	Kymmenkantainen logaritmi

1. JOHDANTO

Yksinkertaisimmillaan verkosto muodostuu, kun eri toimijat vuorovaikuttavat keskenään. Nykyinen yhteiskuntamme sisältää lukuisia erilaisia verkostoja. Näitä ovat esimerkiksi erilaiset logistiikkaverkostot, sähköverkostot, tietoverkot ja mitä moninaisemmat sosiaaliset verkostot. Viime vuosikymmenien teknologinen kehitys on mahdollistanut verkostojen mittavan laajentumisen: internet on paisunut miljardien sivujen verkostoksi, ja erilaiset sosiaaliset mediat ovat räjäyttäneet ihmisten välisten verkostojen laajuuden uudelle tasolle. Samalla kehitys on sallinut entistä tehokkaaman tiedonkeruun näistä verkostoista, ja erilaista dataa on saatavilla valtavia määriä.

Verkostoanalyysillä tarkoitetaan tekniikoita ja menetelmiä, joilla pystytään tutkimaan verkoston ominaisuuksia. Näillä menetelmillä voidaan saada tietoa sekä koko verkoston luonteesta, että myös verkoston yksittäisistä toimijoista. Verkostojen koon ja datan määrän kasvun myötä yhä suuremmaksi ongelmaksi on tullut tunnistaa merkittävimmät ja hyödyllisimmät toimijat verkostoissa. Tässä työssä tutkitaan erilaisia algoritmeja, joilla voidaan määrittää eri toimijoiden arvostus verkostossa. Tärkeimpiä algoritmeja tähän ovat Jon Kleinbergin kehittämä HITS, sekä Larry Pagen ja Sergey Brinin kehittämä PageRank, joka on myös Googlen hakukoneen pohjana. Molemmat näistä algoritmeista on kehitetty alunperin internetin hakukoneita varten, jotta hakutulosten joukosta pystytään erottamaan kaikkein laadukkaimmat ja luotettavimmat lähteet. Näitä algoritmeja voidaan kuitenkin käyttää minkä tahansa verkoston tutkimiseen, ja saada hyödyllistä tietoa näiden toimijoista.

Matemaattisesti verkostoanalyysi pohjautuu erityisesti graafiteoriaan. Graafit tarjoavat hyvän työkalun verkostojen mallintamiseen ja visualisointiin. Graafiteoria kytkeytyy läheisesti matriiseihin, ja matriisit tarjoavat hyvän ympäristön graafien laskennalliselle käsittelylle. Arvostusalgoritmit pohjautuvat graafien rakenteeseen, ja näiden esittäminen graafien matriisien avulla muodostaa erittäin elegantin kokonaisuuden. Näin ollen tämä työ keskittyy teoriaosaltaan graafiteoriaan ja matriisiteoriaan, sekä näiden kahden väliseen yhteyteen. Pohjana graafiteorian osuudelle on erityisesti teos Wasserman ja Faust (1994) *Social network analysis: Methods and applications*. Matriisiteorian osalta tukeudutaan lähinnä teoksiin Meyer (2000) *Matrix analysis and Applied Linear Algebra* sekä Eldén (2007)

Matrix methods in data mining and pattern recognition (Fundamentals of algorithms). Arvostusalgoritmien osalta ensisijaisena lähteenä on Langville ja Meyer (2006) *Google's PageRank and beyond: The science of search engine rankings*, joka tarjoaa myös erittäin kattavan teoreettisen pohjan algoritmien käsittelylle. Lisäksi apuna on käytetty teoksia Ruohonen (2006) *Graafiteoria* sekä Miilumäki (2010) *Web-pohjaisten sosiaalisten verkostojen analyysimenetelmät*.

Esimerkkidatana tässä työssä käytetään Tampereen teknillisen yliopiston kurssien esitietoketjuja, joista muodostuu verkosto. Tämän verkoston ominaisuuksia tutkitaan eri arvostusalgoritmeilla, ja tarkastellaan minkälaisia johtopäätöksiä tästä saatavilla tuloksilla voidaan tehdä.

2. TEORIA

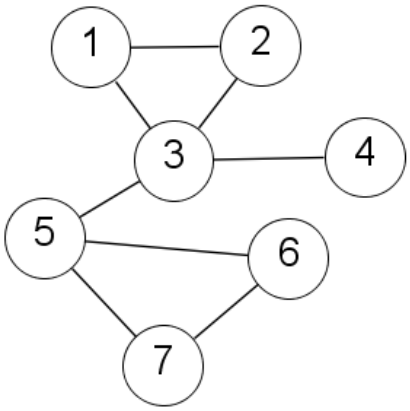
Verkostoanalyysin perustyökalu on graafiteoria. Graafien avulla voidaan mallintaa verkostoja, ja graafiteorian tuloksilla on mahdollista saada tietoa verkoston rakenteesta. Graafiteorian tunnusluvuilla saadaan kuitenkin vain melko triviaalia tietoa verkoston toimijoista. Laskennallisesti graafiteoria kytkeytyy vahvasti matriisilaskentaan, ja tämä tarjoaa formaalin pohjan verkostojen käsittelylle. Verkostojen arvostusalgoritmit pohjautuvatkin lähinnä matriisilaskentaan, ja graafiteoria antaa mallin verkostojen visualisointiin sekä muuttamiseksi matriisimuotoon. Ennen kun voimme perehtyä tarkemmin näihin arvostusalgoritmeihin, tarvitsemme työkaluiksi muutamia graafiteorian, ja hieman laajemman paketin matriisilaskennan tuloksia.

2.1 Graafit

Graafit ovat jo pitkään olleet perustyökalu, joita käytetään verkostojen mallintamisessa. Visuaalisesti graafi koostuu pisteistä, ja näitä pisteitä yhdistävistä viivoista. Havainnollisesti voidaan ajatella, että pisteet kuvaavat verkoston toimijoita, ja viivat kuvaavat näiden toimijoiden välisiä yhteyksiä. Tämä yhteys voi olla suuntaamaton, jolloin yhteys on symmetrinen, tai suunnattu, jolloin yhteydellä on myös suunta. Esimerkkinä suuntaamattomasta yhteydestä voidaan pitää esimerkiksi *'Asuu lähellä'* tai *'On sukua'*. Vastaavasti esimerkkinä suunnatusta yhteydestä voidaan pitää valtioiden välistä tuontia tai vientiä, jolloin tavara siirtyy jostain lähtövaltiosta toiseen valtioon.

Formaalisti graafi $\mathcal{G}$ koostuu kahdesta joukosta: solmujen joukosta $\mathcal{V} = \{n_1, n_2, \ldots, n_g\}$ sekä kaarien joukosta $\mathcal{L} = \{l_1, l_2, \ldots, l_m\}$. Graafissa $\mathcal{G}$ on näin ollen g solmua sekä m kaarta. Graafissa kaaret ovat solmujen pareja $l_k = (n_i, n_j)$, joka kuvaa sitä, että solmujen n_i ja n_j välillä on kaari. Suuntaamattomassa graafissa nämä solmujen muodostamat parit ovat järjestämättömiä, jolloin $(n_i, n_j) = (n_j, n_i)$. Teoriassa graafi voi sisältää myös silmukoita (n_i, n_i), mutta verkostoja tutkiessa oletetaan että toimijat eivät voi olla yhteydessä itseensä. Vastaavasti oletetaan, että solmujen välillä ei voi olla kuin yksi kaari, eli rinnakkaisia kaaria ei sallita. Graafeja, joissa ei ole silmukoita eikä rinnakkaisia kaaria, kutsutaan *yksinkertaisiksi* graafeiksi. Mikäli graafissa ei ole lainkaan kaaria, eli $\mathcal{L} = \{\emptyset\}$, on graafi *tyhjä*. Mikäli graafissa ei ole lainkaan solmuja, eli $\mathcal{L} = \{\emptyset\}$ ja

$\mathcal{V} = \{\emptyset\}$, on graafi ns. *nollagraafi*. Mikäli graafissa on vain yksi solmu, on graafi *triviaali*. Ellei toisin mainita, kaikki tässä työssä käsiteltävät graafit ovat yksinkertaisia.



Kuva 2.1: Esimerkki suuntaamattomasta graafista

Esimerkki 2.1.1. Kuvassa 2.1 on esitetty esimerkki suuntaamattomasta graafista. Tässä graafissa on 7 solmua, jolloin solmujen joukko $\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7\}$, ja 8 kaarta, jolloin kaarien joukko $\mathcal{L} = \{(1, 2), (1, 3), (2, 3), (3, 4), (3, 5), (5, 6), (5, 7), (6, 7)\}$.

Verkostoa, jossa kaikki toimijat ovat samassa roolissa, ja täten solmuilla on samanlaiset ominaisuudet, kutsutaan *yksimoodisiksi*. Mikäli verkosto sisältää kahdenlaisia erilaisia toimijoita, eli verkosto koostuu kahdesta eri solmujoukosta, kutsutaan verkostoa *kaksimoodiseksi*. Tällainen tapaus on esimerkiksi jos ihmiset ovat yhteydessä toisiinsa, muodostaen ensimmäisen solmujoukon, sekä tämän lisäksi yhteydessä yrityksiin, jotka muodostavat oman solmujoukkonsa. Tässä työssä käsiteltävät menetelmät on alunperin suunniteltu yksimoodisten verkostojen analysointiin, ja käytettävä data muodostaa myös yksimoodisen verkoston. Näin ollen tässä työssä perehdytään ainoastaan yksimoodisten verkostojen käsittelyyn, joskin samat konseptit ovat melko helposti yleistettävissä myös kaksimoodisille verkostoille.

2.1.1 Suunnatut graafit

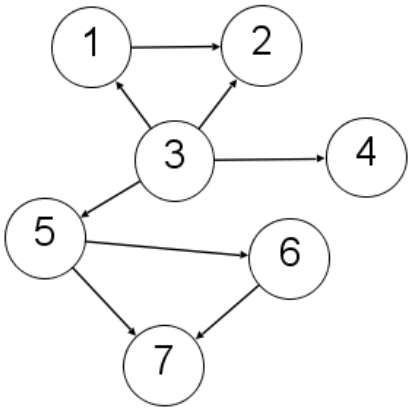
Suuntaamattomat graafit mahdollistavat vain melko triviaalin tieton saamisen verkoston toimijoista. Huomattavasti monipuolisempaa informaatiota on saatavilla, kun toimijoiden välisissä yhteyksissä huomioidaan myös suunta. Suunnatut graafit

eli *digraafit* ovat graafeja, joissa pisteitä yhdistävillä viivoilla on myös suunta. Vastaavasti kuten suuntaamatonkin graafi, digraafi $\mathcal{G}_d$ koostuu pisteiden joukosta $\mathcal{V} = \{n_1, n_2, \ldots, n_g\}$ sekä suunnattujen viivojen eli nuolten joukosta $\mathcal{L} = \{l_1, l_2, \ldots, l_m\}$. Erotuksena suuntaamattomiin graafeihin nuolet muodostuvat järjestetyistä pareista $l_k =< n_i, n_j >$, ja näille yleisesti $< n_i, n_j > \neq < n_j, n_i >$. Nuolessa $< n_i, n_j > n_i$ on *alkupiste* ja n_j *loppupiste*. Solmut voivat näin ollen olla yhteydessä toisiinsa kolmella eri tavalla:

1. Solmujen välillä ei ole nuolta

2. Solmujen välillä on nuoli jompaan kumpaan suuntaan

3. Solmujen välillä on nuoli molempiin suuntiin



Kuva 2.2: Esimerkki suunnatusta graafista

Esimerkki 2.1.2. Kuvassa 2.2 on esitetty esimerkki suunnatusta graafista. Tässä graafissa on 7 solmua, jolloin solmujen joukko $\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7\}$, ja 8 kaarta, jolloin kaarien joukko $\mathcal{L} = \{< 1, 2 >, < 2, 3 >, < 3, 1 >, < 3, 4 >, < 3, 5 >, < 5, 6 > , < 6, 7 >, < 7, 5 >\}$.

2.1.2 Kulku, polku ja graafin yhtenäisyys

Määritelmä 2.1.1. Kulku W on graafin $\mathcal{G}(\mathcal{N}, \mathcal{L})$ solmujen ja kaarien äärellinen jakso

$$W = n_{i0}, l_{j0}, n_{i1}, l_{j1}, \ldots l_{j(k-1)}, n_{ik}$$

missä $n_{i(t-1)}$ ja n_{it} ovat viivan l_{jt} päätepisteet, $t = 0, 1, 2, \ldots$

Kulku alkaa aina solmusta, ja päättyy solmuun. Solmu n_{i0} on kulun alkupiste, ja solmu n_{ik} loppupiste. Kulku W on *reitti*, mikäli jokainen sen sisältämä viiva l_{jt} esiintyy vain kerran. Kulku W on *polku*, mikäli jokainen sen sisältämä piste n_{it} ja viiva l_{jt} esiintyy vain kerran, lukuunottamatta kulkua jossa alku- ja loppupiste ovat samat. Solmu n_i on *saavutettavissa* solmusta n_j, mikäli näiden välillä on polku. Nämä ominaisuudet yleistyvät myös suunnatuille graafeille.

Määritelmä 2.1.2. Suunnattu kulku W_d on digraafin $\mathcal{G}_d(\mathcal{N}, \mathcal{L})$ solmujen ja nuolien äärellinen jakso

$$W_d = n_{i0}, l_{j0}, n_{i1}, l_{j1}, \ldots l_{j(k-1)}, n_{ik}$$

missä $n_{i(t-1)}$ on viivan l_{jt} alkupiste ja *nit* ovat viivan l_{jt} päätepiste

Suunnatussa reitissä jokainen sen sisältämä viiva esiintyy vain kerran, ja suunnatussa polussa jokainen sen sisältämä piste ja viiva esiintyy vain kerran, paitsi jos alku- ja loppupiste ovat samat. Polkujen avulla määritellään graafien yhtenäisyys.

Määritelmä 2.1.3. Graafi on yhtenäinen, jos sen jokaisen solmuparin välillä on polku

Yhtenäisessä graafissa jokainen solmu on saavutettavissa mistä tahansa solmusta. Suunnattujen graafien tapauksessa sanotaan, että graafi on *vahvasti yhtenäinen*, mikäli jokaisen solmuparia yhdistää sekä polku solmusta n_i solmuun n_j, kuin myös polku solmusta n_j solmuun n_i. Vastaavasti digraafi on *heikosti yhtenäinen*, mikäli jokaista solmuparia yhdistää ainakin polku toiseen suuntaan, mutta ei välttämättä molempiin suuntiin. Digraafin yhtenäisyys ja solmujen saavutettavuus ovat keskeisiä ominaisuuksia arvostusmenetelmien laskemiseksi, sillä tällaisen graafin vieruspistematriisi on redusoitumaton.

2.1.3 Graafien tunnusluvut

Yksinkertaisin graafien tunnusluku on solmun aste, joka kuvaa suuntaamattomassa graafissa kuinka monta kaarta lähtee kustakin solmusta.

Määritelmä 2.1.4. Suuntaamattomassa graafissa solmun aste $d(n_i)$ on solmuun n_i liittyneiden kaarien lukumäärä

Toinen tunnusluku on graafin tiheys, joka kuvaa kuinka suuri osa graafin mahdollisista kaarista on olemassa.

Määritelmä 2.1.5. Graafin tiheys Δ on graafin kaarien lukumäärä jaettuna kaikkien mahdollisten kaarien lukumäärällä

Lause 2.1.1. Yksinkertaisen graafin, jossa on g solmua ja m kaarta, tiheydelle Δ pätee

$$\Delta = \frac{2m}{g(g-1)}$$

Yksinkertaisen digraafin, jossa on g solmua ja m nuolta, tiheydelle Δ pätee

$$\Delta = \frac{m}{g(g-1)}$$

Todistus. Yksinkertaisessa graafissa jossa on g solmua on maksimissaan $g(g-1)/2$ kaarta, ja yksinkertaisessa digraafissa jossa on g solmua, on maksimissaan $g(g-1)$ nuolta, joten lause seuraa triviaalisti näistä. □

Graafin tiheys saa arvoja väliltä $[0, 1]$. Mikäli graafi on tyhjä, on sen tiheys 0, ja vastaavasti jos tiheys on 1 eli kaikki mahdolliset kaaret ovat olemassa, on graafi *täydellinen.*

Vastaavasti kuin suuntaamattomalle graafille aste, suunnatulle graafille voidaan määritellä solmun lähtöaste ja tuloaste. Näiden tunnuslukujen avulla voidaan kuvata, kuinka moneen muuhun solmuun kukin solmu on yhteydessä alku- sekä loppupisteen roolissa.

Määritelmä 2.1.6. Digraafissa solmun n_i tuloaste $d_I(n_i)$ on niiden nuolien l_i lukumäärä, joille solmu n_i on loppupiste.

Määritelmä 2.1.7. Digraafissa solmun n_i lähtöaste $d_O(n_i)$ on niiden nuolien l_i lukumäärä, joille solmu n_i on alkupiste.

Määritelmä 2.1.8. Mikäli solmun lähtöaste on nolla ja tuloaste erisuuri kuin nolla, sanotaan solmua nieluksi. Mikäli solmun tuloaste on nolla ja lähtöaste erisuuri kuin nolla, sanotaan solmua lähteeksi.

Esimerkki 2.1.3. Kuvan 2.2 suunnatulle graafille voidaan laskea edellä mainitut tunnusluvut. Solmujen asteet, tuloasteet ja lähtöasteet on esitetty taulukossa 2.1.Graafin tiheys on lauseen 2.1.1 mukaisesti $\Delta = \frac{8}{7*(7-1)} = 0,19$, eli 19% kaikista mahdollisista nuolista on olemassa.

Taulukko 2.1: Esimerkkigraafin solmujen asteet, tuloasteet ja lähtöasteet

Solmu	$d(n_i)$	$d_I(n_i)$	$d_O(n_i)$
1	2	1	1
2	2	2	0
3	4	0	4
4	1	1	0
5	3	2	1
6	2	1	1
7	2	2	0

Esimerkki 2.1.4. Solmu 3 on lähde, koska sen tuloaste on nolla ja lähtöaste erisuuri kuin nolla. Solmut 2, 4 ja 7 ovat nieluja, koska niiden lähtöaste on nolla ja tuloaste erisuuri kuin nolla.

Edellä mainitut solmujen ja kaarien tai nuolien lukumäärää perustuvat tunnusluvut antavat ainoastaan melko triviaalia tietoa graafin rakenteesta. Seuraavat tunnusluvut perustuvat graafin solmujen välisiin polkuihin, ja nämä sisältävät hieman kuvaavampaa informaatiota graafin rakenteesta.

Määritelmä 2.1.9. Geodeesi on lyhin polku graafin kahden solmun välillä.

Määritelmä 2.1.10. Kahden solmun välinen etäisyys $d(i, j)$ määritellään solmujen geodeesin pituutena.

Mikäli solmujen välillä ei ole polkua, niin kyseisten solmujen välinen etäisyys on ääretön. Suuntaamattomilla graafeilla $d(i, j) = d(j, i)$, mutta suunnatuilla graafeilla tämä ei luonnollisestikaan päde.

Määritelmä 2.1.11. Graafin halkaisija $d(\mathcal{G})$ on graafin solmujen välisten etäisyyksien maksimi.

Määritelmä 2.1.12. Solmun n_i eksentrisyys eli epäkeskeisyys on suurin etäisyys solmun n_i ja minkä tahansa graafin muun solmun n_j kanssa.

Määritelmä 2.1.13. Graafin säde $r(\mathcal{G})$ on graafin solmujen eksentrisyyden minimi.

Halkaisija, eksentrisyys ja säde ovat määritettävissä ainoastaan yhtenäiselle graafille. Jos solmujen, joiden välillä ei ole polkua, välinen etäisyys määritellään äärettöman sijasta määrittelemättömäksi, voidaan kuitenkin graafin halkaisija laskea myös graafille joka ei ole yhtenäinen. Tällöin tätä tulkittaessa täytyy kuitenkin ottaa huomioon, että tämä luku ei anna täydellistä informaatiota graafin rakenteesta.

Esimerkki 2.1.5. Kuvassa 2.1 esitetyn graafin halkaisija $d(\mathcal{G}) = 3$, ja säde $r(\mathcal{G}) = 2$. Kuvassa 2.2 esitetyn digraafin halkaisija on $d(\mathcal{G}) = 3$, ja säde määrittelemätön.

Yksi tunnusluku, joka kertoo jo hieman enemmän graafin rakenteesta, on modulaarisuus. Tämä saa arvoja väliltä $[-1, 1]$, ja kuvaa graafin klusterirakennetta. Eri klusterit sisältävät tiheästi toisiinsa yhteydessä olevia solmuja, ja nämä klusterit ovat heikosti yhdistettyjä muihin klustereihin. Mitä korkeampi on graafin modulaarisuus, sitä selkeämmistä klustereista se koostuu. Tällaisen klusterointiongelman ratkaisu on optimointiongelma, joka suuren verkoston tapauksessa on tyypillisesti mahdotonta laskea. Tämän ratkaisemiseksi on kuitenkin olemassa approksimatiivisia algoritmeja, joista yksi löytyy julkaisusta Blondel (2008). Tämä algoritmi on myös käytössä Gephi-ohjelmistossa. Kun solmut on jaettu klustereihin, voidaan graafin modulaarisuus laskea helposti.

Määritelmä 2.1.14. Graafin modulaarisuus Q voidaan laskea yhtälöstä

$$Q = \frac{1}{2n} \sum_{i,j} \left[L_{ij} - \frac{d(n_i)d(n_j)}{2n} \right] \delta(c_i, c_j)$$

Missä c_i on klusteri johon solmu n_i on asetettu, ja funktio

$$\delta(c_i, c_j) = \begin{cases} 1 & \text{jos } i = j \\ 0 & \text{muulloin} \end{cases}$$

Käytännössä tämä mittaa kaarien tai nuolien määrää klustereiden sisällä verrattuna kaarien tai nuolien määrään klustereiden välillä.

2.1.4 Graafien visualisointi

Graafien visualisointiin on lukuisa määrä työkaluja, ja erilaisia ladonta-algoritmeja. Erityisen suosittuja ovat erilaiset voimiin perustuvat algoritmit, jotka yksinkertaisuudessaan toimivat siten, että solmut jotka ovat yhteydessä toisiinsa vetävät toisiaan puoleensa, ja solmut jotka eivät ole yhteydessä toisiinsa hylkivät toisiaan. Näin muodostuvasta visualisoinnista tulee hyvinkin intuitiivinen rakenteeltaan, ja graafin rakenteesta saadaan hyvä yleiskäsitys jo ensimmäisellä silmäyksellä. Ongelmana näissä on helposti laskennan hitaus, ja suurien graafien piirtäminen tällaisella algoritmilla saattaa kestää huomattavan kauan aikaa.

Tässä työssä graafien visualisointiin on käytetty pääasiassa Gephi-ohjelmistoa, joka tarjoaa laajan kirjon erilaisia menetelmiä graafien piirtämiseen, ja myöskin valmiita työkaluja verkostoanalyysiä varten. Ensisijaisena ladonta-algoritmina on käytetty ForceAtlas2-algoritmia, joka perustuu solmujen välisiin voimiin ja gravitaatioon. Algoritmin yksityiskohtia ei ole järkevää käydä tässä läpi, mutta yksityiskohtainen kuvaus tämän toiminnasta on saatavilla Gephin kehittäjien alustavasta julkaisusta *ForceAtlas2, A Graph Layout Algorithm for Handy Network Visualization* [9].

2.1.5 Graafien matriisiesitys

Matriisit tarjoavat elegantin ja yksinkertaisen työkalun graafien käsittelyyn. Yksinkertaisin tällainen matriisi on vieruspistematriisi L, jonka alkiot ovat binäärisiä, ja kuvaavat onko kahden solmun välillä kaari. Tämä antaa pohjan graafien matemaattiselle käsittelylle, ja arvostusalgoritmien lähtökohtana on usein graafin vieruspistematriisi.

Määritelmä 2.1.15. Vieruspistematriisin L alkiot l_{ij} määritellään

$$l_{ij} = \begin{cases} 1 & \text{jos } l_k = (n_i, n_j) \text{ on olemassa} \\ 0 & \text{jos } l_k = (n_i, n_j) \text{ ei ole olemassa} \end{cases}$$

Tämä sama määritelmä pätee sekä suuntaamattomille että suunnatuille graafeille. Suunnatuille graafeille matriisi L on symmetrinen, koska kun $(n_i, n_j) = (n_j, n_i)$, on määritelmän mukaan myös $l_{ij} = l_{ji}$.

Esimerkki 2.1.6. Kuvan 2.2 suunnatun graafin vieruspistematriisi L on

$$L = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Vieruspistematriisin hyödyllinen ominaisuus on että sen potenssit kertovat graafien solmujen välillä olevien kulkujen lukumäärän.

Lause 2.1.2. Vieruspistematriisin L potenssimatriisin L^k alkio $[L^k]_{ij}$ kertoo, kuinka monta kulkua joiden pituus on k, solmujen n_i ja n_j välillä on

Joissain tilanteissa on käytännöllisempää käyttää standardoitua vieruspistematriisia, eli niin sanottua linkkimatriisia H. Tämä määritellään kuten vieruspistematriisi, mutta jokainen rivin standardointiin käytetään kyseisestä solmusta lähtevien nuolien lukumäärää, eli lähtöastetta $d_O(n_i)$. Tällöin jokaisen rivin, jota vastaavasta solmusta lähtee nuolia, summaksi tulee 1. Mikäli rivi vastaa solmua, joka on nielu, on tämän rivin kaikki alkiot nollia. Tämän muotoista matriisia kutsutaan alistokastiseksi matriisiksi, ja sen alkiot voidaan tulkita todennäköisyyksiksi.

Määritelmä 2.1.16. Linkkimatriisin H alkiot h_{ij} määritellään

$$h_{ij} = \begin{cases} \frac{1}{d_O(n_i)} & \text{jos } l_k = (n_i, n_j) \text{ on olemassa} \\ 0 & \text{jos } l_k = (n_i, n_j) \text{ ei ole olemassa} \end{cases}$$

Mikäli graafissa ei ole nieluja, on linkkimatriisin H jokaisen rivin summa 1, jolloin matriisia kutsutaan stokastiseksi matriisiksi. Stokastisten matriisien ominaisuuksiin palataan tässä työssä myöhemmin.

Esimerkki 2.1.7. Kuvan 2.2 suunnatun graafin linkkimatriisi H on

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Koska toisen, neljännen ja seitsemännen solmun tuloaste on nolla, on linkkimatriisin vastaavat rivit nollarivejä, joten tässä tapauksessa linkkimatriisi H ei ole stokastinen.

2.2 Matriisien ominaisuuksia

Jotta graafien matriiseja voidaan hyödyntää arvostusalgoritmien määrittelyssä, on paikallaan käydä läpi muutamia matriisien perusominaisuuksia. Graafien matriisit ovat täysin tavallisia matriiseja, joille pätevät yleiset matriisien laskusäännöt. Matriisien yhteenlasku määritellään matriisien alkioiden yhteenlaskuna, edellyttäen että matriisien dimensiot ovat samat.

Määritelmä 2.2.1. Matriisien peruslaskutoimitukset
$n \times m$ Matriisien A ja B summan $A + B$ alkiot määritellään

$$(A + B)_{ij} = a_{ij} + b_{ij}$$

$n \times m$ Matriisin A ja $m \times n$ matriisin B tulon AB alkiot määritellään

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$n \times m$ Matriisin A ja $m \times 1$ vektorin x tulon Ax alkiot määritellään

$$(Ax)_i = \sum_{k=1}^{n} a_{ik} x_k$$

$n \times n$ Neliömatriisi A on ei-singulaarinen, jos on olemassa $n \times n$ matriisi A^{-1}, jolle

$$AA^{-1} = I = A^{-1}A$$

Tällöin matriisia A^{-1} kutsutaan matriisin A käänteismatriisiksi.
$n \times n$ Matriisin A eksponenttimatriisi e^A määritellään sarjana

$$e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots + \frac{A^k}{k!} + \ldots$$

2.2.1 Matriisinormit

Normi on funktio, joka määrittelee jokaiselle vektoriavaruuden vektorille ei-negatiivisen pituuden. Eri normeja käytetään aina kunkin sovellusalueen mukaan, mutta yleisimmillään normi määritellään seuraavasti.

Määritelmä 2.2.2. Vektorin p-normi määritellään yhtälöllä

$$||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}$$

missä $p \geq 1$

Määritelmästä seuraa, että vektorinormi on aina positiivinen, paitsi nollavektorille nolla. Yleensä vektorin pituuden mittaamiseen käytetään euklidista normia, eli 2-normia

$$||x||_2 = \sqrt{(\sum_{i=1}^{n} |x_i|^2)} \tag{2.2.1}$$

Kuitenkin tarkasteltaessa stokastisia matriiseja ja Markovin ketjuja, on käytännöllisempää käyttää 1-normia

$$||x||_1 = \sum_{i=1}^{n} |x_i| \tag{2.2.2}$$

Vastaavasti joissain tilanteissa on tarpeen käyttää ∞-normia

$$||x||_\infty = \max_i |x_i| \tag{2.2.3}$$

Matriiseille normit määritellään vektorinormien avulla, jolloin sanotaan että matriisinormi on kyseessä olevan vektorinormin indusoima. Kunkin edellämainitun vektorinormin indusoima matriisinormi saa yksinkertaisen muodon, jotka ovat käytännöllisiä matriisien ominaisuuksia tulkittaessa.

Määritelmä 2.2.3. Vektorinormin indusoima matriisinormi määritellään

$$||A|| = \max_{||x||=1} ||Ax||$$

missä $||x||$ on vektorinormi.

Matriisinormit saavat kunkin normin tapauksessa varsin yksinkertaisen muodon, ja näistä kukin antaa hyödyllistä informaatiota matriisin ominaisuuksista.

Lause 2.2.1. 1-normin indusoima matriisinormi on matriisin A suurin itseisarvojen sarakesumma.

$$||A||_1 = \max_j \sum_i |a_{ij}|$$

2-normin indusoima matriisinormi on matriisin $A^T A$ suurimman ominaisarvon λ_{max} neliöjuuri

$$||A||_2 = \sqrt{\lambda_{\max}}$$

∞-normin indusoima matriisinormi on matriisin A suurin itseisarvojen rivisumma

$$||A||_\infty = \max_i \sum_j |a_{ij}|$$

Jokainen matriisinormeista on yhteensopiva indusoivan vektorinorminsa kanssa, toisin sanoen nämä toteuttavat yhtälön

$$||Ax|| \leq ||A||||x||$$

Näiden todistukset ovat melko pitkiä ja vaativat varsinkin 2-normin osalta hieman laajempaa matriisiteorian pohjaa, joten todistukset sivuutetaan. Todistukset ovat löydettävissä esimerkiksi teoksesta Meyer (2000).

2.2.2 Ominaisarvot ja ominaisvektorit

Yksi käyttökelpoisimmista ominaisuuksista on matriisin ominaisarvo, sekä tätä vastaava ominaisvektori. Nämä määritellään siten, että ominaisvektorin kertominen matriisilla ei muuta vektorin suuntaa, vaan ainoastaan skaalaa vektorin pituutta ominaisarvon suuruisesti.

Määritelmä 2.2.4. Skalaari λ sekä vektori $x \neq 0$ ovat matriisin A ominaisarvo sekä tätä ominaisarvoa vastaava (oikeanpuoleinen) ominaisvektori, jos ne toteuttavat yhtälön

$$Ax = \lambda x$$

Jokaista ominaisarvoa vastaa myös vasemmanpuoleinen ominaisvektori y^T, joka määritellään vastaavasti kuin oikeanpuoleinen ominaisvektori

Määritelmä 2.2.5. Ominaisarvoa λ vastaava vasemmanpuoleinen ominaisvektori $y^T \neq 0$ toteuttaa yhtälön

$$y^T A = \lambda y^T$$

Matriisin A ominaisarvot ovat karakteristisen polynomin $p(\lambda) = det(A - \lambda I)$ juuria. Koska karakteristisen polynomin $p(\lambda)$ aste on n, on matriisilla A yhteensä n kappaletta ominaisarvoja. Osa näistä ominaisarvoista voi olla kompleksisia. Mikäli matriisi A on reaalinen, niin kompleksisten ominaisarvojen konjugaatit ovat myös matriisin A ominaisarvoja. Erillisten ominaisarvojen joukkoa $\sigma(A)$ sanotaan matriisin A spektriksi. Matriisin spektrisäde $\rho(A)$ määritellään suurimman ominaisarvon itseisarvona.

Määritelmä 2.2.6. Matriisin spektrisäde

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$$

Kompleksitason ympyrää, jonka keskipisteenä on origo ja säteenä matriisin A spektrisäde $\rho(A)$, kutsutaan spektrikehäksi. Riippuen matriisista tällä kehällä saattaa olla yksi tai useampia ominaisarvoja. Spektrisäde on aina pienempi tai yhtäsuuri kuin mikä tahansa matriisin indusoitu normi.

Lause 2.2.2. Matriisin A spektrisäteelle pätee kaikilla matriisinormeilla

$$\rho(A) \leq ||A||$$

Todistus. Olkoon λ matriisin A ominaisarvo ja x tätä vastaava ominaisvektori. Tällöin

$$|\lambda|||x|| = ||\lambda x|| = ||Ax|| \leq ||A||||x||$$

Ja koska x on ominaisvektori ja siten aina $x \neq 0$, niin jokaisella ominaisarvolla λ pätee

$$|\lambda| \leq ||A||$$

Eli

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| \leq ||A||$$

□

Matriisin A ominaisarvon λ algebrallinen kertaluku $algmult_A(\lambda)$ kuvaa kuinka monikertainen juuri λ on karakteristisessa polynomissa. Mikäli $algmult_A(\lambda) = 1$, sanotaan ominaisarvoa λ yksinkertaiseksi ominaisarvoksi. Ominaisarvon λ geometrinen kertaluku on ominaisarvoa λ vastaavien lineaarisesti riippumattomien ominaisvektorien lukumäärä. Geometriselle kertaluvulle pätee aina $geomult_A(\lambda) \leq algmult_A(\lambda)$. Mikäli $geomult_A(\lambda) = algmult_A(\lambda)$, sanotaan ominaisarvoa λ semiyksinkertaiseksi.

2.2.3 Erityisiä matriiseja

Määritelmä 2.2.7. Permutaatiomatriisi P on matriisi, joka saadaan vaihtamalla identiteettimatriisin I rivien paikkoja, jolloin jokaisella rivillä ja jokaisessa sarakkeessa on tasan yksi ykkönen, ja muut alkiot ovat nollia. Tätä rivien (tai sarakkeiden) vaihdettua järjestystä kutsutaan permutaatioksi.

Lause 2.2.3. Permutaatiomatriisi toteuttaa seuraavat ominaisuudet:

1. Matriisin A kertominen permutaatiomatriisilla P vasemmalta vaihtaa matriisin A rivien järjestystä permutaation mukaisesti

2. Matriisin A kertominen permutaatiomatriisilla P oikealta vaihtaa matriisin A sarakkeiden järjestystä permutaation mukaisesti

3. Permutaatiomatriisin P transpoosi P^T on myös permutaatiomatriisi

Arvostuksien laskemiseksi edellytetään usein että graafia kuvaava matriisi on redusoitumaton. Graafin kannalta tämä tarkoittaa sitä, että graafi on vahvasti yhtenäinen. Tällöin mitä tahansa graafin solmuparia (n_i, n_j) yhdistää polku solmusta n_i solmuun n_j sekä polku solmusta n_i solmuun n_j.

Määritelmä 2.2.8. Matriisi A on redusoituva, jos on olemassa permutaatiomatriisi P, matriisi Y sekä neliömatriisit X ja Z, joille pätee

$$P^T AP = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}$$

Muussa tapauksessa matriisi A on redusoitumaton.

Mikäli matriisit A ja B ovat similaarisia, on niillä useita vastaavia ominaisuuksia. Niiden ominaisarvot ovat samat, mutta ominaisvektorit eivät välttämättä ole samat.

Määritelmä 2.2.9. Matriisit A ja B ovat similaarisia, jos on olemassa ei-singulaarinen matriisi Q, jolle

$$B = QAQ^{-1}$$

Mikäli matriisi A on similaarinen diagonaalimatriisin D kanssa, sanotaan matriisia A diagonalisoituvaksi.

Määritelmä 2.2.10. Matriisi A jonka spektri on $\sigma(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$ on diagonalisoituva, mikäli on olemassa ei-singulaarinen matriisi Q ja diagonaalimatriisi D, joille

$$A = QDQ^{-1}$$

Koska diagonaalimatriisin ominaisarvot ovat sen diagonaalialkioita, nähdään suoraan että matriisi D muodostuu matriisin A ominaisarvoista.

$$D = \begin{pmatrix} \lambda_1 I_{m_1} & 0 & \ldots & 0 \\ 0 & \lambda_2 I_{m_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_s I_{m_s} \end{pmatrix} \tag{2.2.4}$$

Tässä identiteettimatriisien I_{m_i}, $i = 1, 2, \ldots, s$ koko riippuu siitä, kuinka moninkertainen tätä vastaava ominaisarvo λ_i on. Tällöin

$$m_i = algmult_A(\lambda_i) \tag{2.2.5}$$

Diagonalisoituvuudesta seuraa, että matriisit Q ja Q^{-1} sisältävät matriisin A oikean- ja vasemmanpuoleiset ominaisvektorit.

Lause 2.2.4. Diagonalisoituvan matriisin $A = QDQ^{-1}$ oikeanpuoleiset ominaisvektorit muodostavat matriisin Q sarakkeet ja vasemmanpuoleiset ominaisvektorit matriisin Q^{-1} rivit

Todistus. Kirjoitetaan matriisit $Q = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix}$, $Q^{-1} = \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix}$. Yhtälö $A = QDQ^{-1}$ on yhtäpitävä yhtälön $AQ = QD$ kanssa. Tällöin

$$AQ = QD$$

$$A \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}$$

$$\begin{pmatrix} Ax_1 & Ax_2 & \dots & Ax_n \end{pmatrix} = \begin{pmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \dots & \lambda_n x_n \end{pmatrix}$$

Vastaavasti Yhtälö $A = QDQ^{-1}$ on yhtäpitävä yhtälön $Q^{-1}A = DQ^{-1}$ kanssa. Tällöin

$$Q^{-1}A = DQ^{-1}$$

$$\begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix} A = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix} \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix}$$

$$\begin{pmatrix} y_1^T A \\ y_2^T A \\ \vdots \\ y_n^T A \end{pmatrix} = \begin{pmatrix} \lambda_1 y_1^T \\ \lambda_2 y_2^T \\ \vdots \\ \lambda_n y_n^T \end{pmatrix}$$

□

Diagonalisoituvat matriisit voidaan esittää spektrihajotelmana, joka tarjoaa hyödyllisiä ominaisuuksia näiden matriisien käsittelyyn.

Lause 2.2.5. Diagonalisoituva matriisi A jonka spektri on $\sigma(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$ voidaan esittää spektrihajotelmana

$$A = \lambda_1 G_1 + \lambda_2 G_2 + \cdots + \lambda_s Gs$$

Todistus.

$$\begin{aligned}
A &= QDQ^{-1} \\
&= \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{pmatrix} \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix} \\
&= \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1 y_1^T \\ \lambda_2 y_2^T \\ \vdots \\ \lambda_n y_n^T \end{pmatrix} \\
&= \lambda_1 x_1 y_1^T + \lambda_2 x_2 y_2^T + \cdots + \lambda_n x_n y_n^T \\
&= \lambda_1 G_1 + \lambda_2 G_2 + \cdots + \lambda_s Gs
\end{aligned}$$

□

Spektrihajotelman yksi hyvä ominaisuus on, että se mahdollistaa matriisin A funktioiden kirjoittamisen spektrihajotelmana.

Lause 2.2.6. Jos matriisi A on diagonalisoituva, voidaan funktio $f(A)$ kirjoittaa spektrihajotelmana

$$f(A) = f(\lambda_1) G_1 + f(\lambda_2) G_2 + \cdots + f(\lambda_s) G_s$$

Tässä matriisit G_i ovat spektriprojektiomatriiseja, joille pätee seuraavat ominaisuudet:

Lause 2.2.7.

1. $G_i = G_i^2$ on projektiomatriisi ominaisavaruuteen $N(A - \lambda_i I)$
2. $G_1 + G_2 + \cdots + G_s = I$
3. $G_i G_j = 0,$ kun $i \neq j$

4. $G_i = \prod_{\substack{j=1 \\ j \neq i}}^{s} (A - \lambda_j I) / \prod_{\substack{j=1 \\ j \neq i}}^{s} (\lambda_i - \lambda_j), i = 1, 2, \ldots, s$
5. Mikäli λ_i on yksinkertainen ominaisarvo, niin

$$G_i = \frac{x_i y_i^*}{y_i^* x_i}$$

missä x_i ja y_i^* ovat ominaisarvoa λ_i vastaavat oikean- ja vasemmanpuoleinen ominaisvektori

Täydellinen todistus näille lauseille löytyy esimerkiksi luentomonisteesta Smith (2007), kappale 15.

Määritelmä 2.2.11. Rivistokastinen matriisi on matriisi, jonka jokaisen rivin summa on 1, ja sarakestokastinen matriisi on matriisi, jonka jokaisen sarakkeen summa on 1

Käyttötarkoituksesta ja määrittelyistä riippuen eri tilanteissa on tarpeen käyttää joko rivi-, tai sarakestokastisia matriiseja. Tässä työssä määritelmät on tehty siten, että useimmissa tapauksissa käytetään rivistokastisia matriiseja. Ellei toisin mainita, termillä "stokastinen matriisi"tarkoitetaan rivistokastista matriisia.

2.2.4 Kertomenetelmä

Kertomenetelmä on iteratiivinen menetelmä, jolla saadaan ratkaistua diagonalisoituvan matriisin A suurin ominaisarvo λ_1 ja tätä vastaava ominaisvektori x_1. Tämä menetelmä on tärkeässä roolissa eri arvostusalgoritmien laskemisessa. Olkoon matriisin A ominaisarvot

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_k| \tag{2.2.6}$$

Oletus $|\lambda_1| > |\lambda_2|$ implikoi, että λ_1 on reaalinen, koska muutoin myös $\overline{\lambda}_1$ olisi matriisin A ominaisarvo jolla olisi sama itseisarvo kuin ominaisarvolla λ_1. Olkoon funktio $f(z) = (z/\lambda_1)^n$, ja tiedetään, että $|\lambda_i/\lambda_1| < 1$ kaikilla $i = 2, 3, \ldots, k$ jolloin käyttämällä matriisin A spektrihajotelmaa 2.2.5 ja 2.2.6 saadaan

$$\begin{aligned}\left(\frac{A}{\lambda_1}\right)^n &= f(A) = f(\lambda_1)G_1 + f(\lambda_2)G_2 + \cdots + f(\lambda_k)G_k \\ &= G_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^n G_2 + \cdots + \left(\frac{\lambda_k}{\lambda_1}\right)^n G_k \xrightarrow{n\to\infty} G_1\end{aligned}$$

Näin ollen jokaiselle vektorille x_0 jolle $G_1 x_0 \neq 0$ pätee $(A^n x_o/\lambda_1^n) \rightarrow G_1 x_o \in N(A - \lambda_1 I)$, eli $(A^n x_o/\lambda_1^n)$ suppenee kohti suurinta ominaisarvoa λ_1 vastaavaa ominaisvektoria x. Huomioitavaa tässä on, että jakaja λ_1^n on skalaari, joten vektori $A^n x_o$ kääntyy kohti vektorin x suuntaa kun $n \rightarrow \infty$. Koska yhtälön 2.2.2 mukaan kaikki matriisinormit ovat suurempia kuin spektrisäde eli suurimman ominaisarvon itseisarvo, voidaan skaalaamiseen käyttää mitä tahansa matriisinormia. Myöhemmin todetaan, että stokastisille matriiseille $\lambda_1 = 1$, jolloin kertomenetelmä supistuu muotoon

$$x_{n+1} = Ax_n \tag{2.2.7}$$

Vastaavasti voidaan osoittaa, että kertomenetelmä toimii myös vasemmanpuoleisille ominaisvektoreille, jolloin stokastisille matriiseille pätee

$$y_{n+1}^T = y_n^T A \tag{2.2.8}$$

Yhtälön 2.2.7 perusteella voidaan todeta että menetelmän suppenemisnopeus riippuu siitä, kuinka nopeasti $(\lambda_2/\lambda_1)^n \rightarrow 0$. Myöhemmin nähdään että Pagerankin yhteydessä ominaisarvoon λ_2 voidaan vaikuttaa parametrin α valinnalla, jolloin voidaan kontrolloida kuinka monta iteraatiota tarvitaan halutun tarkkuuden saavuttamiseksi.

2.2.5 Perron-Frobenius-teoreema

Matriisin A sanotaan olevan ei-negatiivinen, jos sen kaikki arvot $a_{ij} \geq 0$. Tämä voidaan ilmaista merkinnällä $A \geq 0$. Vastaavasti matriisin A sanotaan olevan positiivinen, jos sen kaikki arvot $a_{ij} > 0$, jolloin käytetään merkintää $A > 0$. Verkostojen matriisit ovat selkeästi ei-negatiivisia. Näiden matriisien ominaisarvoille ja ominaisvektoreille on olemassa monia hyödyllisiä ominaisuuksia, jotka todisti ensimmäisen kerran positiivisille matriiseille Oskar Perron vuonna 1907, ja jonka työtä täydensi ei-negatiivisille matriiseille Georg Frobenius vuonna 1912.

Perronin teoreeman mukaan positiivisille matriiseille pätee useita hyödyllisiä ominaisuuksia. Näistä tärkein on suurimman ominaisarvon yksinkertaisuus, mikä takaa kertomenetelmän suppenemisen kohti yksikäsitteistä suurinta ominaisarvoa vastaavaa ominaisvektoria.

Lause 2.2.8. Perronin teoreema
Positiiviselle matriisille A, jonka spektrisäde $r = \rho(A)$, pätee seuraavat ominaisuudet

1. $r > 0$

2. $r \in \sigma(A)$
3. $algmult_A(r) = 1$, missä ominaisarvoa r kutsutaan Perronin juureksi
4. On olemassa positiivinen ominaisvektori $x > 0$ jolle $Ax = rx$ ja positiivinen vasemmanpuoleinen ominaisvektori $y^T > 0$, jolle $y^T A = ry^T$
5. Perronin vektori on yksikäsitteinen positiivinen vektori, joka määritellään $Ap = rp$, $p > 0$, $||p||_1 = 1$
   ja A:lla ei ole muita positiivisia ominaisvektoreita, riippumatta ominaisarvosta
6. Vastaavilla oletuksilla on olemassa myös vasemmanpuoleinen Perronin vektori q^T jolle pätee
   $q^T A = rq^T, q > 0, ||q||_1 = 1$
7. r on ainoa ominaisarvo matriisin A spektrikehällä
8. Collatz-Wielandt-yhtälö on voimassa, jonka mukaan
   $r = \max_{x \in \mathcal{N}} f(x)$
   missä
   $f(x) = \min_{\substack{1 \le i \le n \\ x_i \neq 0}} \frac{[Ax]_i}{x_i}$ ja $\mathcal{N} = \{x | x \geq 0 \text{ ja } x \neq 0\}$

Näistä ominaisuuksista useimmat kuitenkin menetetään, jos A on ei-negatiivinen. Frobenius kuitenkin huomasi että tässä merkittävää ei sinällään ole nollien olemassaolo matriisissa A, vaan se missä paikassa nämä nollat sijaitsevat. Ratkaisevaksi tässä muodostuu se, että nollat sijaitsevat matriisissa A siten, että matriisi A on redusoitumaton.

Lause 2.2.9. Perron-Frobenius-teoreema
Redusoitumattomalle ei-negatiivisille matriisille A, jonka spektrisäde on $r = \rho(A)$ pätee seuraavat ominaisuudet

1. $r > 0$
2. $r \in \sigma(A)$
3. $algmult_A(r) = 1$, missä ominaisarvoa r kutsutaan Perronin juureksi
4. On olemassa positiivinen ominaisvektori $x > 0$ jolle $Ax = rx$ ja positiivinen vasemmanpuoleinen ominaisvektori $y^T > 0$, jolle $y^T A = ry^T$

5. Perronin vektori on yksikäsitteinen positiivinen vektori, joka määritellään
   $Ap = rp, p > 0, ||p||_1 = 1$
   ja A:lla ei ole muita positiivisia ominaisvektoreita, riippumatta ominaisarvosta

6. Vastaavilla oletuksilla on olemassa myös vasemmanpuoleinen Perronin vektori q^T jolle pätee
   $q^T A = rq^T, q > 0, ||q||_1 = 1$

7. r ei välttämättä ole ainoa ominaisarvo matriisin A spektrikehällä

8. Collatz-Wielandt-yhtälö on voimassa, jonka mukaan
   $r = \max_{x \in \mathcal{N}} f(x)$
   missä
   $f(x) = \min_{\substack{1 \leq i \leq n \\ x_i \neq 0}} \frac{[Ax]_i}{x_i}$ ja $\mathcal{N} = \{x | x \geq 0 \text{ ja } x \neq 0\}$

Näin ollen ainoastaan ominaisuus numero 7 on erilainen kuin positiivisille matriiseille. Todistukset sekä Perronin teoreemalle, että Perron-Frobenius-teoreemalle löytyvät esimerkiksi teoksesta Meyer (2000), kappale 8.

Mikäli redusoitumattomalla ei-negatiivisella matriisilla A on ainoastaan yksi ominaisarvo r spektrikehällä, sanotaan matriisia A primitiiviseksi. Ainoastaan tällöin matriisin A normitetuilla potensseilla on raja-arvo, ja tämä on ratkaiseva tekijä sille, suppeneeko kertomenetelmä kohti yksikäsitteistä suurinta ominaisvektoria.

Lause 2.2.10. Redusoitumaton ei-negatiivinen matriisi A, jonka spektrisäde on $r = \rho(A)$, on primitiivinen jos ja vain jos raja-arvo $\lim_{k \to \infty}(A/r)^k$ on olemassa, jolloin

$$\lim_{k \to \infty} \left(\frac{A}{r} \right)^k = \frac{pq^T}{q^T p} > 0$$

missä p ja q^T ovat matriisin A oikean- ja vasemmanpuoleiset Perronin vektorit.

Primitiivisyys voidaan todeta helposti kahdella yksinkertaisella testillä.

Lause 2.2.11. Ei-negatiiviselle neliömatriisille A pätee seuraavat ominaisuudet:

- A on primitiivinen, jos se on redusoitumaton ja sillä on ainakin yksi nollasta poikkeava diagonaalialkio.
- A on primitiivinen, jos ja vain jos $A^k > 0$ jollain $m > 0$.

Todistukset näille löytyvät teoksesta Meyer (2000), kappale 8.

2.3 Markovin ketjut

Stokastinen prosessi määritellään joukkona satunnaismuuttujia $\{X_t\}_{t=0}^{\infty}$, joilla on sama arvojoukko $\{S_1, S_2, \ldots, S_n\}$, jota kutsutaan prosessin tila-avaruudeksi. Parametri t mielletään useimmiten ajaksi, ja X_t kuvaa prosessin tilaa ajanhetkellä t. Aika oletetaan tässä diskreetiksi, ja tila-avaruus äärelliseksi. Markovin ketju on stokastinen prosessi, joka toteuttaa jokaisella $t = 0, 1, 2, \ldots$ Markovin ehdon

$$P(X_{t+1} = S_j | X_t = S_{i_t}, X_{t-1} = S_{i_{t-1}}, \ldots, X_0 = S_{i_0}) = P(X_{t+1} = S_j | X_t = S_{i_t}) \tag{2.3.1}$$

Tässä $P(E|F)$ tarkoittaa ehdollista todennäköisyyttä. Markovin ehdosta seuraa, että ketjun seuraava tila riippuu ainoastaan ketjun nykyisestä tilasta, eikä lainkaan edeltävistä tiloista. Näin ollen prosessia sanotaan muistittomaksi.

Markovin ketjun kuvaamiseen käytetään usein suunnattua graafia, jossa nuolet kuvaavat siirtymiä tilojen välillä. Markovin ketjua jossa siirtymätodennäköisyydet eivät riipu ajasta, kutsutaan stationääriseksi Markovin ketjuksi. Tällaisen ketjun siirtymätodennäköisyysmatriisi $P = [p_{ij}]$ kuvaa todennäköisyyksiä siirtyä tilasta i tilaan j. Jotta kyseessä olisi Markovin ketju, on matriisin P oltava stokastinen. Stokastiselle matriisille P jokaisen rivin summa on 1, joten

$$Pe = e \tag{2.3.2}$$

Näin ollen $\lambda = 1$ on stokastisen matriisin P ominaisarvo, jota vastaa ominaisvektori e. Koska stokastiselle matriisille P pätee myös, että $||P||_\infty = 1$, yhtälöiden 2.2.2 ja 2.2.6 myötä matriisin P spektrisäde on 1, ja näin ollen suurin ominaisarvo $\lambda_1 = 1$.

Tilajakauma on rivivektori $p^T = (p_1, p_2, \ldots, p_n)$, jolle pätee $||p||_1 = 1$. Näin ollen stokastisen matriisin P jokainen rivi on tilajakauma. Stationäärinen tilajakauma Markovin ketjulle, jonka siirtymätodennäköisyysmatriisi on P, on tilajakauma π^T joka toteuttaa ehdon

$$\pi^T P = \pi^T \tag{2.3.3}$$

Mille tahansa alkutilalle $p^T(0)$ voidaan laskea k:nnen askeleen tilajakauma

$$p^T(k) = p^T(0)P^k \tag{2.3.4}$$

Tämä voidaan katsoa erityistapaukseksi vasemmanpuoleisen ominaisvektorin kertomenetelmälle. Tietynlaisille Markovin ketjuille tämä iteraatio suppenee kohti yksikäsitteistä stationääristä tilajakaumaa.

Redusoitumaton Markovin ketju on ketju, jonka siirtymätodennäköisyysmatriisi P on redusoitumaton. Vastaavasti Markovin ketju on primitiivinen, jos sen siirtymätodennäköisyysmatriisi P on primitiivinen. Ketjulle joka on redusoitumaton sekä primitiivinen, on olemassa yksikäsitteinen stationäärinen tilajakauma, jota kohti ketju suppenee.

Lause 2.3.1. Redusoitumaton ja primitiivinen Markovin ketju, jonka siirtymätodennäköisyysmatriisi on P, suppenee kohti staattista tilajakaumaa π^T

Todistus. Koska Markovin ketjun siirtymätodennäköisyysmatriisille $Pe = e$, on matriisin P oikeanpuoleinen Perronin vektori e/n. Vastaavasti staattinen tilajakauma π^T toteuttaa ehdon $\pi^T P = \pi^T$, jolloin se on matriisin P vasemmanpuoleinen Perronin vektori. Tällöin yhtälön 2.2.10 perusteella siirtymätodennäköisyysmatriisin P raja-arvo on

$$\lim_{k\to\infty} = \frac{(e/n)\pi^T}{\pi^T(e/n)} = \frac{e\pi^T}{\pi^T e} = e\pi^T = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_n \\ \pi_1 & \pi_2 & \dots & \pi_n \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_n \end{pmatrix} > 0$$

Tällöin minkä tahansa tilajakauman raja-arvo on

$$\lim_{k\to\infty} p^T(k) = \lim_{k\to\infty} p^T(0)P^k = p^T(0)e\pi^T = \pi^T$$

□

Tästä nähdään että ketjun suppeneminen ei riipu alkutilasta $p^T(0)$, mikä on varsin käyttökelpoinen ominaisuus.

3. ARVOSTUSALGORITMIT

Arvostusalgoritmit ovat tyypillisesti saaneet alkunsa internetin hakukoneiden kehityksen yhteydessä. Hakukoneiden toimintaan liittyy kaksi perustavanlaatuista ongelmaa:

1. Miten löytää hakua vastaavat objektit?

2. Miten löytää laajasta hakutulosjoukosta ne tulokset, jotka ovat laadukkaimpia?

Ensimmäisen kysymyksen käsittely sisältää lukuisia mielenkiintoisia ongelmia muun muassa semantiikan alueella, mutta tämän käsittely jätetään tämän työn ulkopuolelle. Ensimmäiset ratkaisut toiseen kysymykseen vastaamiseksi perustuivat jonkin asiantuntijan tekemään arvioon kyseisen sivun laadusta. Internetin kasvaessa räjähdysmäisesti oli selvää, että tällaisella menetelmällä ei ollut mahdollista pysytellä kehityksessä mukana. Merkittävämpi edistysaskel arvostusten määrittelyssä saatiin, kun keksittiin internetin linkkirakenteen mukaisesti määräytyvät arvostukset. Nämä tarjosivat mahdollisuuden suurien verkostojen käsittelyyn, ja matemaattisen formalismin joka takasi objektiiviset arvostukset. Tässä esitellään näistä algoritmeista kaksi merkittävintä, Pagerank ja HITS, joihin liittyvät konseptit ovat enemmän tai vähemmän kaikkien verkoston linkkirakenteisiin perustuvien arvostusalgoritmien pohjana.

3.1 Pagerank

Pagerank on Googlen perustajien Larry Pagen ja Sergey Brinin kehittämä arvostusalgoritmi, jolla annetaan jokaiselle internetin sivulle yksi sivun laatua kuvaava arvo. Algoritmi perustuu koko internetin linkkirakenteeseen, ja ottaa arvostuksia laskiessa kaikki internetin indeksoidut sivut huomioon. Näin ollen Pagerank-arvot päivittyvät uudelleen aina kun indeksointi suoritetaan uudelleen, mikä tarkoitti aiemmin että arvostukset saattoivat olla jopa kuukauden vanhoja. Nykyisin indeksointimenetelmätkin ovat kehittyneet siinä määrin, että indeksointi sekä Pagerank-arvot päivittyvät lähes reaaliajassa. Pagerank perustuu ajatukseen, että sivun arvo määräytyy sen mukaan, kuinka arvokkaat sivut linkittävät tälle

sivustolle. Tämä voidaan formuloida matemaattisesti niin sanotun Googlematriisin ominaisarvoyhtälönä. Vaikka Pagerank onkin alunperin suunniteltu internetsivujen arvostukseen, voidaan sitä käyttää minkä tahansa suunnatun verkon toimijoiden arvostamiseen.

3.1.1 Alkuperäinen määritelmä

Olkoon n_i joku verkon solmu, ja kaikki verkoston solmut indeksoitu $i = 1 \dots g$. Olkoon tällöin O_i niiden verkoston solmujen joukko, jotka ovat alkupisteenä sellaisille nuolille, joille n_i on loppupiste. Olkoon vastaavasti I_i niiden solmujen joukko, jotka ovat loppupisteenä sellaisille nuolille, joille n_i on alkupiste. Tällöin solmusta n_i lähtevien nuolien lukumäärä on solmun lähtöaste $d_O(n_i)$. Solmun n_i Pagerank määräytyy sen mukaan, kuinka korkea Pagerank on solmuilla, jotka kuuluvat joukkoon I_i. Näin ollen solmun n_i Pagerank r_i määritellään joukon I_i solmujen Pagerank arvojen painotettuna summana, jossa painokertoimena on kullakin joukon I_i solmulla kyseisen solmun lähtöaste.

$$r_i = \sum_{j \in I_i} \frac{r_j}{d_O(n_j)} \tag{3.1.1}$$

Tämä alustava määritelmä on rekursiivinen, joten arvoja ei voida laskea suoraan. Sen sijaan voidaan ottaa alkuarvaus r^0, jonka jälkeen iteroidaan yhtälöä

$$r_i^{(k+1)} = \sum_{j \in I_i} \frac{r_j^{(k)}}{d_O(n_j)}, k = 0, 1, \dots \tag{3.1.2}$$

Yhtälöt 3.1.1 ja 3.1.2 laskevat ainoastaan yhden toimijan Pagerank arvon kerrallaan. Huomattavasti yksinkertaisempaa on käyttää matriiseja, jolloin vektori r sisältää kaikkien verkoston toimijoiden Pagerank arvot. Käyttämällä hyväksi linkkimatriisin H määritelmää 2.1.16 voidaan yhtälö 3.1.1 kirjoittaa muodossa

$$r^T = r^T H \tag{3.1.3}$$

Tämä voidaan tulkita niin, että r on linkkimatriisin H ominaisarvoa $\lambda = 1$ vastaava vasemmanpuoleinen ominaisvektori. Vastaavasti yhtälö 3.1.2 voidaan kirjottaa muodossa

$$r^{(k+1)T} = r^{(k)T} H, k = 0, 1, \dots \tag{3.1.4}$$

Tässä iteroinnissa on kuitenkin se ongelma, että mikäli joku solmu on nielu, kerää tämä nielu arvostusta, mutta tämä arvostus ei jakaudu eteenpäin, eikä näinollen iterointi suppene. Myöskään ei ole itsestään selvää, että $\lambda = 1$ on edes matriisin H yksinkertainen ominaisarvo. Näin ollen matriisiin H tarvitsee tehdä muutoksia,

jotta laskeminen onnistuu. Brinin ja Pagen ratkaisu tähän ongelmaan oli matriisin H muokkaaminen stationäärisen Markovin ketjun siirtymätodennäköisyysmatriisiksi, jolle Markovin ketjujen teorian myötä on useita hyödyllisiä ominaisuuksia.

3.1.2 Googlematriisin muodostaminen

Koska verkoston linkkimatriisi H saattaa sisältää nollarivejä, ei se välttämättä ole stokastinen, eikä näin ollen staattisen Markovin ketjun siirtymätodennäköisyysmatriisi. Brinin ja Pagen keino muuttaa H stokastiseksi oli niin kutsuttu satunnaisen surffaajan malli. Tämä tulee ajatuksesta, jonka mukaan internetissä satunnainen verkon selaaja etenee verkossa linkkien mukaan, kunnes saapuu nieluun. Nieluun saapuessa satunnainen selaaja siirtyy mihin tahansa verkon solmuun yhtä suurella todennäköisyydellä, joka saadaan aikaan lisäämällä jokaisesta nielusta nuoli kaikkiin muihin verkoston solmuihin. Näin saadaan aikaan stokastinen linkkimatriisi S.

Määritelmä 3.1.1. Stokastinen linkkimatriisi

$$S = H + a(\frac{1}{n}e^T)$$

missä a on nieluvektori, jonka alkiot a_i määritellään

$$a_i = \begin{cases} 1 & \text{jos } n_i \text{ on nielu} \\ 0 & \text{muulloin} \end{cases}$$

ja e on vektori, jonka kaikki alkiot ovat ykkösiä

Matriisi S on selvästi stokastinen, koska jokaisen rivin summa on 1. Tämä ei kuitenkaan riitä vielä takaamaan, että matriisia S vastaava Markovin ketju suppenisi kertomenetelmällä iteroitaessa. Tämän vuoksi matriisiin S sovelletaan uudelleen satunnaisen surffaajan mallia. Tässä tapauksessa ajatuksena on, että satunnainen selaaja saattaa koska tahansa kyllästyä etenemään linkkien mukaan, ja sen sijaan siirtyä satunnaisesti johonkin muuhun verkon solmuun. Tätä kuvaa teleportaatiomatriisi $\frac{1}{n}ee^T$, joka kuvaa että jokaisesta solmusta on yhtä suuri todennäköisyys siirtyä mihin tahansa verkoston solmuun. Ottamalla stokastisesta linkkimatriisista S ja tästä teleportaatiomatriisista konveksi summa, saadaan tulokseksi Googlematriisi

Määritelmä 3.1.2. Googlematriisi

$$G = \alpha S + (1 - \alpha)\frac{1}{n}ee^T$$

Tässä parametri α on vapaasti valittavissa, ja tällä voidaan säädellä kuinka suuri painoarvo annetaan verkoston rakenteelle ja kuinka suuri satunnaiselle siirtymiselle verkostossa. Tämä myös vaikuttaa suoraan Googlematriisin toiseen ominaisarvoon λ_2, joka määrää kuinka nopeasti kertomenetelmä suppenee.

Esimerkki 3.1.1. Esimerkin 2.1.7 linkkimatriisi H muokataan stokastiseksi linkkimatriisiksi S seuraavasti

$$\begin{aligned} S = H + a(\frac{1}{n}e^T) &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{7}\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{pmatrix} \end{aligned}$$

Esimerkki 3.1.2. Edellisen esimerkin stokastinen linkkimatriisi S muokataan edelleen Googlematriisiksi G. Käytetään esimerkiksi parametria $\alpha = 0,9$.

$$\begin{aligned} G &= \alpha S + (1-\alpha)\frac{1}{n}ee^T \\ &= 0,9 * \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{pmatrix} + \frac{0,1}{7} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0,014 & 0,914 & 0,014 & 0,014 & 0,014 & 0,014 & 0,014 \\ 0,143 & 0,143 & 0,143 & 0,143 & 0,143 & 0,143 & 0,143 \\ 0,239 & 0,239 & 0,014 & 0,239 & 0,239 & 0,014 & 0,014 \\ 0,143 & 0,143 & 0,143 & 0,143 & 0,143 & 0,143 & 0,143 \\ 0,014 & 0,014 & 0,014 & 0,014 & 0,014 & 0,464 & 0,464 \\ 0,014 & 0,014 & 0,014 & 0,014 & 0,014 & 0,014 & 0,914 \\ 0,143 & 0,143 & 0,143 & 0,143 & 0,143 & 0,143 & 0,143 \end{pmatrix} \end{aligned}$$

Lause 3.1.1. Jos stokastisen matriisin S spektri on $\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$, niin Googlematriisin $G = \alpha S + (1-\alpha)\frac{1}{n}ee^T$ spektri on $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$.

Todistus. Koska S on stokastinen, on 1 sen ominaisarvo ja e tätä vastaava ominaisvektori. Olkoon $Q = \begin{pmatrix} e & X \end{pmatrix}$ ei-singulaarinen matriisi jolla on matriisin S ominaisvektori e ensimmäisenä sarakkeena. Olkoon $Q^{-1} = \begin{pmatrix} y^T \\ Y^T \end{pmatrix}$. Tällöin

$$Q^{-1}Q = \begin{pmatrix} y^Te & y^TX \\ Y^Te & Y^TX \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I \end{pmatrix}$$

josta nähdään että $y^Te = 1$ ja $Y^Te = 0$. Kun suoritetaan similaarisuusmuunnos matriisille S

$$Q^{-1}SQ = \begin{pmatrix} y^Te & y^TSX \\ Y^Te & Y^TSX \end{pmatrix} = \begin{pmatrix} 1 & y^TSX \\ 0 & Y^TSX \end{pmatrix}$$

nähdään että matriisi Y^TSX sisältää loput matriisin S ominaisarvot $\lambda_2, \ldots, \lambda_n$. Kun suoritetaan similaarisuusmuunnos matriisille $G = \alpha S + (1-\alpha)\frac{1}{n}ee^T$, saadaan

$$\begin{aligned} Q^{-1}(\alpha S + (1-\alpha)\frac{1}{n}ee^T)Q &= \alpha Q^{-1}SQ + \frac{(1-\alpha)}{n}Q^{-1}ee^TQ \\ &= \begin{pmatrix} \alpha & \alpha y^TSX \\ 0 & \alpha Y^TSX \end{pmatrix} + \frac{(1-\alpha)}{n}\begin{pmatrix} y^Te \\ Y^Te \end{pmatrix}\begin{pmatrix} e^Te & e^TX \end{pmatrix} \\ &= \begin{pmatrix} \alpha & \alpha y^TSX \\ 0 & \alpha Y^TSX \end{pmatrix} + \begin{pmatrix} \frac{(1-\alpha)}{n} & \frac{(1-\alpha)}{n}e^TX \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & \alpha y^TSX + \frac{(1-\alpha)}{n}e^TX \\ 0 & \alpha Y^TSX \end{pmatrix} \end{aligned}$$

Näin ollen Googlematriisin $G = \alpha S + (1-\alpha)\frac{1}{n}ee^T$ ominaisarvot ovat $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots \alpha\lambda_n\}$. □

Koska matriisin S suurin ominaisarvo on $\lambda_1 = 1$, on selvää että Googlematriisin toiseksi suurin ominaisarvo on korkeintaan α. Mitä laajempi verkosto on kyseessä, sitä todennäköisempää on, että $\lambda = 1$ on useampikertainen ominaisarvo. Tämän vuoksi useimmiten Pagerankia sovellettaessa oletetaan, että $\lambda_2 = 1$, ja kertomenetelmän suppenemisnopeus riippuu siitä, kuinka nopeasti $\alpha^k \to 0$. Google käyttää tiettävästi arvoa $\alpha \approx 0,85$, ja noin 50 iteraatiota, jolloin päästään noin 2-3 desimaalin tarkkuuteen.

Esimerkki 3.1.3. Esimerkkien 3.1.1 ja 3.1.2 stokastisen linkkimatriisin S ja Googlematriisin G ominaisarvot ovat esitettynä taulukossa 3.1. Tuloksista nähdään, että Googlematriisin G ominaisarvot todella olevat lauseen 3.1.1 mukaiset. Nähdään myös, että stokastinen linkkimatriisi S on primitiivinen, koska se on redusoitumaton ja sillä on vain yksi ominaisarvo jolle $|\lambda| = 1$. Vastaavasti myös Googlematriisi G on primitiivinen, koska se on redusoitumaton ja sillä on vain yksi ominaisarvo jolla $|\lambda| = 1$.

3.1.3 Lopullinen määritelmä

Googlematriisi G on selvästi stokastinen, koska se on kahden stokastisen matriisin konveksi summa. Matriisia G vastaavassa graafissa jokainen solmu on yhteydessä toisiinsa, jolloin graafi on triviaalisti vahvasti yhtenäinen ja siten matriisi G redusoitumaton. Koska matriisin kaikki alkiot ovat nollaa suurempia, on matriisi

Taulukko 3.1: Esimerkkigraafin stokastisen linkkimatriisin S sekä Googlematriisin G ominaisarvot

i	$\lambda_i(S)$	$\lambda_i(G)$	$\lvert\lambda_i(S)\rvert$	$\lvert\lambda_i(G)\rvert$
1	1	1	1	1
2	-0,1,4+0,2041i	-0,126+0,1837i	0,54	0,49
3	-0,1,4-0,2041i	-0,126-0,1837i	0,54	0,49
4	-0,2915	-0,2624	0,25	0,22
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0

myöskin primitiivinen. Tällöin matriisi G kuvaa redusoitumatonta primitiivistä Markovin ketjua, joka suppenee kertomenetelmällä iteroitaessa kohti yksikäsitteistä stationääristä tilajakaumaa. Näin ollen Pagerank määritellään Googlematriisin G ominaisarvoa $\lambda_1 = 1$ vastaavana vasemmanpuoleisena ominaisvektorina, eli Googlematriisia G vastaavan Markovin ketjun stationäärisenä tilajakaumana.

Määritelmä 3.1.3. Pagerank r^T on vektori, joka toteuttaa yhtälön

$$r^T G = r^T$$

Ja tämä voidaan ratkaista iteratiivisesti kertomenetelmällä käyttäen mitä tahansa alkutilaa $r^T(0)$

$$r^T(k+1) = r^T(k)G = r^T(0)G^k \tag{3.1.5}$$

Koska matriisin G kaikki alkiot ovat nollasta poikkeavia, voi tämä iterointi olla erittäin raskas suuria verkostoja käsiteltäessä. Tällöin on järkevää hajottaa matriisi G määritelmänsä mukaisesti tekijöihin, jolloin iteraatio saa muodon

$$\begin{aligned} r^T(k+1) &= r^T(k)\left(\alpha S + \frac{1-\alpha}{n} e e^T\right) \\ &= \alpha r^T(k)S + \frac{1-\alpha}{n} r^T(k) e e^T \\ &= \alpha r^T(k)H + \frac{(\alpha r^T(k)a + 1 - \alpha)}{n} e^T \end{aligned} \tag{3.1.6}$$

Näin ollen kullakin iteraatiolla suoritetaan vain vektori-matriisikertolasku $r^T(k)H$ sekä vektori-vektorikertolasku $r^T(k)a$. H on useimmiten erittäin harva matriisi, jossa useimmat alkiot ovat nollia. Näin ollen jokaisessa iteraatiossa tarvitsee suorittaa

huomattavasti vähemmän laskutoimituksia, kuin jos iterointi suoritettaisiin suoraan matriisiin G.

Esimerkki 3.1.4. Kuvassa 2.2 esitetyn esimerkkigraafin solmujen Pagerank-arvot parametrilla $\alpha = 0,9$ on esitetty taulukossa 3.2. Arvokkaimmiksi solmuiksi nousevat solmut 2 ja 7, jotka ovat myös solmut joilla on korkeimmat tuloasteet. Näistä solmu 7 on arvokkaampi, koska siihen osoittavat solmut 5 ja 6 ovat arvokkaampia kuin solmuun 2 osoittavat solmut 1 ja 3. Solmut 1, 4 ja 5 saavat kaikki saman arvostuksen, koska jokaiseen näistä osoittaa ainoastaan solmu 3. Kaikkein vähiten arvostettu solmu on solmu 3, joka on graafin ainoa lähde. Näin ollen nämä tulokset havainnollistavat hyvin Pagerankin perusajatusta.

Taulukko 3.2: Esimerkkigraafin solmujen Pagerank-arvot parametrilla $\alpha = 0,9$

Solmu	Pagerank
1	0,107
2	0,202
3	0,087
4	0,107
5	0,107
6	0,135
7	0,256

3.1.4 Personoitu Pagerank

Pagerankin määritelmässä käytetty ajatus satunnaissiirtymä, jota kuvattiin teleportaatiomatriisilla $\frac{1}{n}ee^T$, on lopulta melko karkea yksinkertaistus todellisesta tilanteesta. On varsin luonnollista ajatella, että internetiä selaava käyttäjä ei etene täysin sattumanvaraisesti sivulta toiselle, vaan tämän liikkumista ohjaavat jonkinlaiset käyttäjäkohtaiset mieltymykset. Nämä mieltymykset voidaan ottaa huomioon määrittelemällä personointivektori, jonka avulla määritellään personoitu teleportaatiomatriisi, joka kuvaa todennäköisyyksiä miten käyttäjä siirtyy verkostossa silloin, kun hän ei etene verkoston linkkirakenteen mukaisesti.

Määritelmä 3.1.4. Personointivektori v on vektori, jonka pituus $||v||_1 = 1$, ja jonka alkiot kuvaavat todennäköisyyksiä joilla käyttäjä siirtyy verkostossa alkiota vastaavaan solmuun satunnaissiirtymän tapauksessa

Näin ollen personoitu Googlematriisi saa seuraavanlaisen muodon

Määritelmä 3.1.5. Personoitu Googlematriisi

$$G_p = \alpha S + (1 - \alpha)ev^T$$

Tälle matriisille pätevät samat ominaisuudet kun alkuperäisellekin Googlematriisille. Personoitu Pagerank määritellään vastaavasti kuin normaali Pagerank tämän matriisin suurinta ominaisarvoa $\alpha = 1$ vastaavana vasemmanpuoleisena ominaisvektorina.

Määritelmä 3.1.6. Personoitu Pagerank

$$r_p^T G_p = r_p^T$$

Ongelmaksi personoidun Pagerankin tapauksessa tulee laskenta-aika. Siinä missä alkuperäisen Pagerankin hienous oli siinä, että se laskettiin kerralla koko verkostolle, tarvii personoidussa tapauksessa laskea erillinen vektori jokaiselle käyttäjälle erikseen. Suurempien verkostojen ja käyttäjämäärien tapauksessa tämä tekee täysin mahdottomaksi personoidun Pagerankin soveltamisen. Esimerkiksi Googlen tapauksessa koko verkon Pagerank-arvojen laskeminen vie huomattavan paljon aikaa, ja on selvää ettei tätä ole mitään mahdollisuutta suorittaa miljardeille käyttäjille erikseen. Personinti on näin ollen tällä hetkellä sovellettavissa ainoastaan huomattavasti pienemmän kokoluokan ongelmiin, mutta tietokoneiden laskentatehon kasvaessa on mahdollista että tämä joskus tulee perusominaisuudeksi hakukoneissa.

Toinen ongelma on relevantin personoinnin muodostaminen. Jotta hyvä personointivektori voidaan muodostaa, tarvitaan kattavaa tietoa käyttäjän mieltymyksistä ja liikkumisesta verkostossa. Googlen tapauksessa tätä tietoa on luonnollisesti saatavilla hyvinkin paljon, mutta toisaalta seuraavaksi ongelmaksi muodostuu että personointiprosessi ruokkii itseään. Kun esimerkiksi käyttäjän hakutuloksissa painottuvat aina hänen useasti käyttämät sivustonsa, nousee näiden arvostus kyseiselle käyttäjälle jatkuvasti, eikä hän enää saa niin hyvin tietoa hakutuloksissa harvemmin vierailluilta sivustoilta. Näin ollen Pagerankin alkuperäinen ajatus hämärtyy, koska arvostus ei enää olekaan objektiivinen näkemys kyseisen sivuston arvostuksesta.

Esimerkki 3.1.5. Jos oletetaan, että käyttäjä siirtyy verkostossa 5 kertaa todennäköisemmin solmuun 1 kuin muihin solmuihin, on personointivektori v^T tällöin muotoa

$$v^T = \frac{1}{11}\begin{pmatrix}5 & 1 & 1 & 1 & 1 & 1 & 1\end{pmatrix}$$

Näin ollen personoitu Googlematriisi G_p parametrilla $\alpha = 0,9$ saa muodon

$$\begin{aligned}
G_p &= \alpha S + (1-\alpha)ev^T \\
&= 0,9 * \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\
\frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\
\frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7}
\end{pmatrix} + \frac{0,1}{11}\begin{pmatrix}
5 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix} \\
&= \begin{pmatrix}
0,045 & 0,909 & 0,009 & 0,009 & 0,009 & 0,009 & 0,009 \\
0,174 & 0,138 & 0,138 & 0,138 & 0,138 & 0,138 & 0,138 \\
0,270 & 0,234 & 0,009 & 0,234 & 0,234 & 0,009 & 0,009 \\
0,174 & 0,138 & 0,138 & 0,138 & 0,138 & 0,138 & 0,138 \\
0,045 & 0,009 & 0,009 & 0,009 & 0,009 & 0,459 & 0,459 \\
0,045 & 0,009 & 0,009 & 0,009 & 0,009 & 0,009 & 0,909 \\
0,174 & 0,138 & 0,138 & 0,138 & 0,138 & 0,138 & 0,138
\end{pmatrix}
\end{aligned}$$

Taulukossa 3.3 on esitetetty solmujen personoidut Pagerank-arvot, kun personointi on edellä mainittu painotus. Koska satunnaissiirtymän todennäköisyys solmuun 1 on korkeampi kuin muihin solmuihin, nousee sen arvostus huomattavasti. Vastaavasti myöskin solmun 2 arvostus nousee, koska tämä on ainoa solmu johon solmusta 1 lähtee nuoli. Kaikkien muiden solmujen arvostus laskee, jotta kaikkien arvostusten summa on edelleen yksi.

3.1.5 CheiRank

Cheirank on Pagerankin jatke, joka määritellään kuten Pagerank, mutta verkostolle jonka nuolien suunta on käännetty. Käytännössä tämä tarkoittaa sitä, että Cheirankilla arvokkaimmat solmut ovat niitä, joista lähtee paljon nuolia. Tämä antaa mahdollisuudet tarkastella solmun arvoa kahdessa ulottuvuudessa, vastaavaan tapaan kuin myöhemmin esitettävällä HITS-algoritmilla. Laskennallisesti Cheirank voidaan määrittää vastaavasti kuten Pagerank, mutta

Taulukko 3.3: Esimerkkigraafin solmujen personoidut Pagerank-arvot parametrilla $\alpha = 0,9$

Solmu	Pagerank
1	0,274
2	0,310
3	0,052
4	0,064
5	0,064
6	0,081
7	0,154

nuolien suunnan muuttuessa täytyy graafin matriisit määritellä uudelleen. Tähän tarkoitukseen otamme käyttöön käännetyn linkkimatriisin H_c, stokastisen linkkimatriisin S_c sekä Googlematriisin G_c. Näille pohjana on käännetyn verkoston vieruspistematriisi L_c, joka on yksinkertaisesti alkuperäisen verkoston vieruspistematriisin L transpoosi

Määritelmä 3.1.7. Verkoston, jonka nuolien suunta on käännetty, vieruspistematriisi on alkuperäisen verkoston vieruspistematriisin transpoosi

$$L_c = L^T$$

Määritelmä 3.1.8. Käännetyn linkkimatriisin H_c alkiot $h_{c_{ij}}$ määritellään

$$h_{c_{ij}} = \begin{cases} \frac{1}{d_I(n_i)} & \text{jos } l_k = (n_j, n_i) \text{ on olemassa} \\ 0 & \text{jos } l_k = (n_i, n_j) \text{ ei ole olemassa} \end{cases}$$

Käännetty stokastinen linkkimatriisi määritellään vastaavasti kuin stokastinen linkkimatriisi, sillä erotuksella että vektorina käytetään lähdevektoria b

Määritelmä 3.1.9. Käännetty stokastinen linkkimatriisi

$$S_c = H_c + b(\frac{1}{n}e^T)$$

missä b on lähdevektori, jonka alkiot b_i määritellään

$$b_i = \begin{cases} 1 & \text{jos } n_i \text{ on lähde} \\ 0 & \text{muulloin} \end{cases}$$

Käännetty Googlematriisi määritellään käännetyn stokastisen linkkimatriisin ja teleportaatiomatriisin konveksina summana.

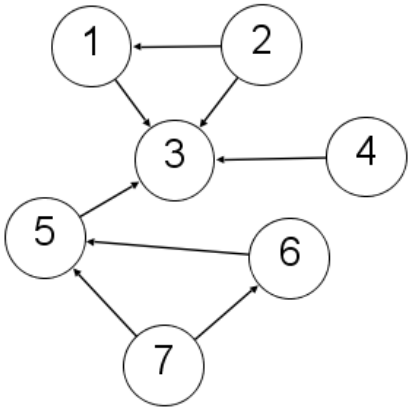
Määritelmä 3.1.10. Käännetty Googlematriisi

$$G_c = \alpha S_c + (1 - \alpha)\frac{1}{n}ee^T$$

Cheirank saadaan laskettua tämän käännetyn Googlematriisin suurinta ominaisarvoa $\lambda_1 = 1$ vastaavana ominaisvektorina c^T, ja tämä voidaan ratkaista kuten Pagerank kertomenetelmällä.

Määritelmä 3.1.11. Cheirank

$$c^T G_c = c^T$$



Kuva 3.1: Esimerkkigraafi, jossa nuolien suunta on käännetty

Esimerkki 3.1.6. Kuvassa 2.2 esitetyn esimerkkigraafin käännetty graafi on esitetty kuvassa 3.1. Käännettyä graafia kuvaavat käännetyt vieruspistematriisi L^T, linkkimatriisi H_c, stokastinen linkkimatriisi S_c ja Googlematriisi G_c ovat seuraavanlaiset:

$$L^T = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad H_c = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

$$S_c = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} \quad G_c = \begin{pmatrix} \frac{1}{70} & \frac{1}{70} & \frac{32}{35} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} \\ \frac{13}{28} & \frac{1}{70} & \frac{13}{28} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{70} & \frac{1}{70} & \frac{32}{35} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} \\ \frac{1}{70} & \frac{1}{70} & \frac{32}{35} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} \\ \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{32}{35} & \frac{1}{70} \\ \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{1}{70} & \frac{13}{28} & \frac{13}{28} \end{pmatrix}$$

Esimerkki 3.1.7. Kuvassa 2.2 esitetyn esimerkkigraafin solmujen Cheirank-arvot parametrilla $\alpha = 0,9$ on esitetty taulukossa 3.4. Ylivoimaisesti arvostetuimmaksi solmuksi nousee solmu 3, josta lähtee kaikkein eniten nuolia. Toiseksi arvostetuin on solmu 5, josta lähtee kaksi nuolta, kolmanneksi arvostetuimmat ovat solmut 1 ja 6 jotka molemmat osoittavat yhteen nieluun, ja vähiten arvostettuja ovat solmut 2,4 ja 7, jotka ovat nieluja. Tämä havainnollistaa selkeästi Cheirankin perusajatusta.

3.2 HITS

HITS, joka on lyhenne sanoista Hypertext Induced Topic Search, on Jon Kleinbergin vuonna 1998 keksimä arvostusalgoritmi, jota Pagerankin tapaan käytettiin alunperin internetin hakukoneiden tulosten arvostamiseen. Pagerankin tavoin HITS perustuu verkoston linkkirakenteeseen, ja vaikka menetelmät kehitettiinkin täysin erillään toisistaan, sisältävät ne monia hyvin samankaltaisia ajatuksia. Suurin ero tulee siitä, että siinä missä Pagerank määritetään koko

Taulukko 3.4: Esimerkkigraafin solmujen Cheirank-arvot parametrilla $\alpha = 0,9$

Solmu	Cheirank
1	0,098
2	0,068
3	0,415
4	0,068
5	0,186
6	0,098
7	0,068

verkostolle, HITS on hakukohtainen ja määritetään haun yhteydessä ainoastaan hakutuloksille. Tämä toisaalta hidastaa yksittäistä hakua, mutta on kuitenkin minimaalinen laskenta-ajaltaan verrattuna koko verkostolle kerralla laskettavaan arvostukseen. HITS voidaan toki laskea myös koko verkostolle, jolloin algoritmin perusajatus on hyvinkin lähellä Pagerankia. Toinen ratkaiseva ero on että HITS antaa kaksi arvoa kullekin solmulle: hubiarvon ja auktoriteettiarvon. Hubi on solmu, josta lähtee paljon nuolia, ja auktoriteetti on solmu johon osoittaa paljon nuolia. HITS-algoritmin ajatus on, että hyviä auktoriteetteja ovat solmut, joihin osoittavat hyvät hubit, ja hyviä hubeja ovat solmut jotka osoittavat hyviin auktoriteetteihin.

HITS-algoritmi alkaa graafin muodostamisella hakutulosten joukosta. Hakutuloksista muodostetaan niin sanottu ympäristögraafi *(neighborhood graph)*, joka määritellään siten, että solmujen välillä on nuoli, mikäli hakutulosten välillä on linkki. Ympäristögraafi $\mathcal{N}$ muodostuu solmujen joukosta $\mathcal{V}$ sekä viivojen joukosta $\mathcal{L}$, joten muodoltaan tämä vastaa siten täysin koko internetin linkkigraafia. Sovellettaessa HITS-menetelmää muualle kuin hakukoneisiin, on usein tarpeen tarkastella koko graafin solmuja ympäristögraafin solmujen sijaan. Laskennallisesti tällä ei sinällään ole merkitystä, vaan menetelmä määritellään samalla tavalla riippumatta tarkastellaanko koko graafia vai jotain tämän osagraafia. Näin ollen tässä työssä tarkastellaan menetelmää koko graafille tapahtuvan laskennan osalta.

HITS voidaan esittää formaalisti kahdella yhtälöllä. Olkoon jokaisella solmulla n_i auktoriteettiarvo x_i ja hubiarvo y_i. Tällöin joillain alkuarvauksilla $x_i^{(0)}$ ja $y_i^{(0)}$ voidaan rekursiivisesti laskea

$$x_i^{(k)} = \sum_{j:(n_j,n_i)\in\mathcal{L}} y_j^{(k-1)} \tag{3.2.1}$$

$$y_i^{(k)} = \sum_{j:(n_i,n_j)\in\mathcal{L}} x_j^{(k)} \tag{3.2.2}$$

Tämä tarkoittaa sitä, että kunkin iteraation auktoriteettiarvo riippuu edellisen iteraation hubiarvoista, ja kunkin iteraation hubiarvo riippuu kyseisen iteraation auktoriteettiarvoista. Vieruspistematriisin L avulla nämä yhtälöt voidaan kirjoittaa muotoon

$$x^{(k)} = L^T y^{(k-1)} \tag{3.2.3}$$

$$y^{(k)} = L x^{(k)} \tag{3.2.4}$$

Sijoittamalla $x^{(k)}$ alempaan yhtälöön, ja $y^{(k-1)}$ ylempään, tämä voidaan taas yksinkertaistaa muotoon

$$x^{(k)} = L^T L x^{(k-1)} \tag{3.2.5}$$

$$y^{(k)} = L L^T y^{(k-1)} \tag{3.2.6}$$

Nämä kaksi yhtälöä vastaavat ominaisvektorin laskemista kertomenetelmällä matriiseille $L^T L$ ja LL^T. Koska matriisi $L^T L$ määrää auktoriteettiarvot, kutsutaan sitä auktoriteettimatriisiksi, ja vastaavasti matriisi LL^T määrää hubiarvot, joten sitä kutsutaan hubimatriisiksi. Molemmat matriisit $L^T L$ ja LL^T ovat symmetrisiä ja positiivisesti semidefiniittejä. Koska molemmat näistä matriiseista ovat reaalisia, tästä seuraa että niiden ominaisarvot ovat reaalisia ja ei-negatiivisia. On kuitenkin mahdollista, että suurin ominaisarvo λ_1 on karakteristisen polynomin moninkertainen juuri, jolloin on mahdollista, että kertomenetelmä suppenee kohti eri vektoreita riippuen alkuarvauksesta. Alkuperäisestä algoritmista onkin kehitetty useita variaatioita, joilla pyritään pääsemään eroon tästä ongelmasta ja parantamaan tuloksia.

Esimerkki 3.2.1. Kuvan 2.2 esimerkkigraafin auktoriteetti-, ja hubimatriisit ovat seuraavanlaiset:

$$LL^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad L^T L = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}$$

Esimerkki 3.2.2. Koska sekä auktoriteetti-, että hubimatriiseilla on molemmilla yksinkertainen suurin ominaisarvo $\lambda_1 = 4,3028$, voidaan esimerkkigraafin arvostusta tutkia alkuperäisellä HITS-algoritmilla. Tällä saadut auktoriteetti- ja hubiarvot on esitetty taulukossa 3.5. Arvostetuimmaksi auktoriteetiksi nousee solmu 2, jonka jälkeen yhtä arvokkaita ovat solmut 1, 4 ja 5. Ylivoimaisesti arvostetuin hubi on solmu 3, ja toiseksi arvostetuin solmu 1. Näiden lisäksi lähteitä ja nieluja lukuunottamatta kaikki solmut saavat jonkun auktoriteetti- tai hubiarvon, mutta nämä jäävät olemattoman pieniksi. Arvostus siis kasautuu tiettyihin kohtiin graafissa, mikä ei ole kovinkaan suotava ominaisuus. Näin ollen alkuperäiseen algoritmiin tarvitaan parannuksia, muutenkin kuin vain yksikäsitteisen suurimman ominaisarvon λ_1 aikaansaamiseksi.

Taulukko 3.5: Esimerkkigraafin solmujen auktoriteetti- ja hubiarvot HITS-algoritmilla

Solmu	Auktoriteetti	Hubi
1	0,232	0,232
2	0,303	0
3	0	0,768
4	0,232	0
5	0,232	$4,16 * 10^{-7}$
6	$2,74 * 10^{-7}$	$2,57 * 10^{-7}$
7	$4,44 * 10^{-7}$	0

3.2.1 Modifioitu HITS

Ensimmäinen ratkaisu arvostusvektorien yksikäsitteisyysongelmaan on vastaava muokkaus kuin Pagerankin tapauksessa. Ongelma poistuu jos matriisit ovat redusoitumattomia, jolloin Perron-Frobenius-teoreeman nojalla on olemassa yksikäsitteinen suurinta ominaisarvoa vastaava ominaisvektori, eli Perronin vektori. Jotta voidaan varmistua matriisien $L^T L$ ja LL^T redusoitumattomuudesta, tehdään niille samankaltainen muutos mitä käytettiin Pagerankin yhteydessä. Tällöin lisätään jokaisesta solmusta nuoli kaikkiin solmuihin, joka automaattisesti takaa että tällaiseen graafiin liittyvä matriisi on redusoitumaton sekä primitiivinen.

$$L^T L \to \xi L^T L + \frac{(1-\xi)}{n} e e^T$$

$$LL^T \to \xi LL^T + \frac{(1-\xi)}{n} e e^T$$

Koska matriisit eivät ole stokastisia, täytyy kertomenetelmässä ottaa huomioon normitus. Vaikka alkuarvauksena voidaankin käyttää mitä tahansa positiivista normalisoitua vektoria, on käytännöllisintä käyttää alkuarvauksena vektoria $x^{(0)} = e/n$. Tällöin saadaan modifioidulle HITS-algoritmille sen lopullinen muoto

$$\begin{aligned} x^{(k)} &= \xi L^T L x^{(k-1)} + \frac{(1-\xi)}{n} e \\ x^{(k)} &= \frac{x^{(k)}}{||x^{(k)}||_1} \end{aligned} \tag{3.2.7}$$

$$\begin{aligned} y^{(k)} &= \xi L L^T y^{(k-1)} + \frac{(1-\xi)}{n} e \\ y^{(k)} &= \frac{y^{(k)}}{||y^{(k)}||_1} \end{aligned} \tag{3.2.8}$$

Esimerkki 3.2.3. Kuvassa 2.2 esitetyn esimerkkigraafin solmujen auktoriteetti- ja hubiarvot modifioidulla HITS-algoritmilla parametrilla $\xi = 0,9$ on esitetty taulukossa 3.6. Arvostetuin auktoriteetti on solmu 2, ja lähes yhtä arvostettuja ovat solmut 1, 4 ja 5, ja näihin kaikkiin osoittaa arvostetuin hubi eli solmu 3. Tämän lisäksi ainoastaan solmu 1 saa merkittävän hubiarvon, ja muut kuin edellä mainitut auktoriteetti- ja hubiarvot jäävät melko mitättömiksi. Näin ollen modifioidun HITS-algoritmin tapauksessa havaitaan vastaavaa arvostuksen kasautumista, mikä oli ongelmana myös alkuperäisen algoritmin käytössä. Tämä asettaa modifioidun HITS-algoritmin käytön kyseenalaiseksi, mikä epäilys vahvistuu myöhemmin algoritmia käytettäessä esimerkkidatan tutkimiseen.

Taulukko 3.6: Esimerkkigraafin solmujen auktoriteetti- ja hubiarvot modifioidulla HITS-algoritmilla parametrilla $\xi = 0,9$

Solmu	Auktoriteetti	Hubi
1	0,228	0,228
2	0,295	0,004
3	0,004	0,743
4	0,228	0,004
5	0,228	0,010
6	0,008	0,008
7	0,010	0,004

3.2.2 Eksponentiaalinen HITS

Toinen esitetty ratkaisu on niin sanottu eksponentiaalinen HITS. Tässä vieruspistematriisi L korvataan matriisilla $e^L - I$. Matriisiteoriasta tiedetään, että matriisieksponentti e^L voidaan kirjoittaa muodossa

$$e^L = I + L + \frac{L^2}{2!} + \frac{L^3}{3!} + \cdots + \frac{L^k}{k!} + \ldots \tag{3.2.9}$$

Tällöin matriisi $e^L - I$ saa muodon

$$e^L - I = L + \frac{L^2}{2!} + \frac{L^3}{3!} + \cdots + \frac{L^k}{k!} + \ldots \tag{3.2.10}$$

Kuten todettiin, vieruspistematriisin L potenssimatriisin L^k alkiot $[L^k]_{ij}$ kuvaavat, kuinka monta kulkua solmujen n_i ja n_j välillä on. Näin ollen matriisi $e^L - I$ voidaan siis tulkita kaikkien näiden kulkujen painotettuna summana, jossa pidempiä kulkuja painotetaan kertoimella $\frac{1}{k!}$, kun kulun pituus on k. Kun matriisi L korvataan matriisilla $e^L - I$, saa exponentiaalinen HITS-algoritmi lopullisen muotonsa

$$\begin{aligned} x^{(k)} &= (e^L - I)^T (e^L - I) x^{(k-1)} \\ x^{(k)} &= \frac{x^{(k)}}{||x^{(k)}||_1} \end{aligned} \tag{3.2.11}$$

$$\begin{aligned} y^{(k)} &= (e^L - I)(e^L - I)^T y^{(k-1)} \\ y^{(k)} &= \frac{y^{(k)}}{||y^{(k)}||_1} \end{aligned} \tag{3.2.12}$$

Tämän menetelmän ongelmana on, että se toimii ainoastaan mikäli tarkasteltava graafi $\mathcal{G}$ on ainakin heikosti yhtenäinen. Tässä työssä käsiteltävästä datasta muodostuva graafi ei ole edes heikosti yhtenäinen, joten tämän menetelmän soveltaminen ei ole mahdollista. Lisäksi matriisieksponenttien laskeminen on hidasta, joten algoritmin soveltaminen suuriin verkostoihin vaatii huomattavan paljon laskentatehoa. Kuvassa 2.2 esitetty esimerkkigraafi on kuitenkin heikosti yhtenäinen, joten tämän puitteissa on mahdollista tarkastella eksponentiaalisen HITS-algoritmin toimintaa.

Esimerkki 3.2.4. Kuvassa 2.2 esitetyn esimerkkigraafin eksponentiaaliset

auktoriteetti- ja hubimatriisit ovat seuraavanlaiset:

$$(e^L - I)^T(e^L - I) = \begin{pmatrix} 1 & 1,5 & 0 & 1 & 1 & 0,5 & 0,67 \\ 1,5 & 3,25 & 0 & 1,5 & 1,5 & 0,75 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1,5 & 0 & 1 & 1 & 0,5 & 0,67 \\ 1 & 1,5 & 0 & 1 & 1 & 0,5 & 0,67 \\ 0,5 & 0,75 & 0 & 0,5 & 0,5 & 1,25 & 1,83 \\ 0,67 & 1 & 0 & 0,67 & 0,67 & 1,83 & 3,69 \end{pmatrix}$$

$$(e^L - I)(e^L - I)^T = \begin{pmatrix} 1 & 0 & 1,5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1,5 & 0 & 5,94 & 0 & 1,5 & 0,67 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1,5 & 0 & 3,25 & 1,5 & 0 \\ 0 & 0 & 0,67 & 0 & 1,5 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Esimerkki 3.2.5. Kuvassa 2.2 esitetyn esimerkkigraafin solmujen auktoriteetti- ja hubiarvot eksponentiaalisella HITS-algoritmilla on esitetty taulukossa 3.7. Huomionarvoista tässä on mahdollisuus, että solmun auktoriteetti- tai hubiarvo voi olla nolla, mikä on syytä ottaa huomioon kun tuloksia hyödynnetään. Arvostetuin auktoriteetti on tässäkin tapauksessa solmu 2, mutta toiseksi arvokkaimmaksi solmuksi nousee solmu 7, jonka auktoriteettiarvo on hyvin lähellä solmua 2. Solmut 1,4 ja 5 saavat saman arvon, solmu 6 hitusen näitä pienemmän ja lähteenä solmu 3 saa auktoriteettiarvon nolla. Arvostetuin hubi on edelleen solmu 3, mutta toiseksi arvostetuimmaksi nousee solmu 5, jonka jälkeen tulevat solmu 1 ja 6. Nämä tulokset ovat varsin erilaiset kuin mitä alkuperäisellä ja modifioidulla HITS-algoritmilla saatiin, joten käytettäessä algoritmin eri variaatioita on ensiarvoisen tärkeää että käyttäjä ymmärtää näiden erot ja heikkoudet.

3.2.3 Satunnaistettu HITS

Kolmas esitetty ratkaisu on niin sanottu satunnaistettu HITS, joka myöskin muistuttaa paljolti Pagerankin määritelmää. Tämä perustuu Pagerankin tapaan ajatukseen, että satunnainen käyttäjä etenee verkostossa linkkirakenteen

Taulukko 3.7: Esimerkkigraafin solmujen auktoriteetti- ja hubiarvot eksponentiaalisella HITS-algoritmilla

Solmu	Auktoriteetti	Hubi
1	0,137	0,126
2	0,239	0
3	0	0,517
4	0,137	0
5	0,137	0,242
6	0,133	0,115
7	0,218	0

mukaisesti. Satunnainen käyttäjä aloittaa satunnaisesta solmusta, ja siirtyy jokaisella askeleella todennäköisyydellä $(1 - \xi)$ uuteen satunnaiseen solmuun. Todennäköisyydellä ξ satunnainen käyttäjä siirtyy joko satunnaisesti valitun lähtevän nuolen suuntaan, tai satunnaisesti valitun tulevan nuolen suuntaan, riippuen siitä onko askel pariton vai parillinen. Näin muodostuu satunnaiskulku, ja parittomista askelista muodostuvan satunnaiskulun stationäärinen tilajakauma kuvaa auktoriteettiarvoja. Vastaavasti parillisista askelista muodostuvan satunnaiskulun stationäärinen tilajakauma kuvaa hubiarvoja. Matemaattisesti tämä voidaan kirjoittaa kahdella yhtälöllä.

$$\begin{aligned} x^{(k)} &= (1-\xi)e + \xi L_{row}^T y^{(k-1)} \\ y^{(k)} &= (1-\xi)e + \xi L_{col} x^{(k)} \end{aligned} \tag{3.2.13}$$

Tässä matriisi L_{row} on vieruspistematriisi L, jonka rivit ovat normalisoitu siten, että niiden summa on 1. Tämä on siten sama kuin aiemmin määritelty linkkimatriisi H. Matriisi L_{col} on vastaavasti vieruspistematriisi L, jonka sarakkeet on normalisoitu siten, että niiden summa on 1. Tämä on sama kuin aiemmin määritelty käännetty linkkimatriisi H_c. Muotoilemalla nämä yhtälöt uudelleen saadaan ne kertomenetelmän edellyttämään muotoon, ja huomioimalla normitus saadaan satunnaistetulle HITS-algoritmille lopullinen muoto

$$\begin{aligned} x^{(k)} &= (1-\xi)e + \xi L_{row}^T((1-\xi)e + \xi L_{col} x^{(k-1)}) \\ x^{(k)} &= \frac{x^{(k)}}{||x^{(k)}||_1} \end{aligned} \tag{3.2.14}$$

$$\begin{aligned} y^{(k)} &= (1-\xi)e + \xi L_{col}((1-\xi)e + \xi L_{row}^T y^{(k-1)}) \\ y^{(k)} &= \frac{y^{(k)}}{||y^{(k)}||_1} \end{aligned} \tag{3.2.15}$$

Esimerkki 3.2.6. Kuvassa 2.2 esitetyn esimerkkigraafin solmujen auktoriteetti- ja hubiarvot satunnaistetulla HITS-algoritmilla parametrilla $\xi = 0,9$ on esitetty taulukossa 3.8. Tuloksista huomataan, että pieniä nyanssieroja lukuunottamatta tulokset ovat hyvinkin samankaltaiset, kuin eksponentiaalisella HITS-algoritmilla. Satunnaistetun HITS-algoritmin tapauksessa arvostusten jakauma on hieman tasaisempi, ja solmun 6 arvostukset nousevat hieman verrattuna eksponentiaaliseen variaatioon. Huomion arvoista on että satunnaistetulla HITS-algoritmilla solmujen auktoriteettiarvojen järjestys on sama kuin solmujen Pagerank-arvojen, ja hubiarvojen sama kuin Cheirank-arvojen. Tämä on luonnollinen seuraus siitä, että algoritmien määritelmät ovat hyvin lähellä toisiaan.

Taulukko 3.8: Esimerkkigraafin solmujen auktoriteetti- ja hubiarvot satunnaistetulla HITS-algoritmilla parametrilla $\xi = 0,9$

Solmu	Auktoriteetti	Hubi
1	0,102	0,148
2	0,223	0,039
3	0,045	0,333
4	0,102	0,039
5	0,102	0,237
6	0,141	0,164
7	0,284	0,039

3.3 Muita algoritmeja

Muut algoritmit voidaan jakaa karkeasti kahteen joukkoon: niihin jotka perustuvat verkoston linkkirakenteeseen, ja niihin, jotka perustuvat solmusta toiseen kulkevien polkujen pituuteen. Varhaisimmat arvostusalgoritmit perustuivat jonkin asiantuntijajoukon subjektiivisesti määrittämiin auktoriteettiarvoihin, ja muita solmuja arvioitiin sen mukaan, kuinka pitkä oli lyhin polku määritellystä auktoriteettisolmusta kyseiseen solmuun. Toinen vaihtoehto oli käyttää asiantuntijoita arvioimaan jokin otos luotettavia solmuja, ja tämän jälkeen koneellisesti etsittiin samankaltaisia luotettavia solmuja automaattisesti verkostosta. Näiden menetelmien heikkoutena oli ensinnäkin tarvittava työmäärä, sillä luonnollisesti verkoston kasvaessa tulee asiantuntijoille mahdottomaksi arvioida kattavaa joukkoa luotettavista solmuista. Toisekseen asiantuntijoiden käyttäminen johti aina subjektiiviseen arvioon kyseisen solmun arvosta, mikä taas

saattoi vaihdella eri asiantuntijoiden välillä, joka taas teki arvioinneista epäluotettavampia.

3.3.1 SALSA

Eräs algoritmi, jonka ajatuksena on yhdistää sekä Pagerankin että HITS-algoritmin parhaat puolet, on *stochastic approach for link structure analysis*, eli SALSA. Tämän kehittivät Ronny Lempel ja Shlomo Moran vuonna 2000. Algoritmin ajatuksena on yhdistää Pagerankin Markovin ketjuihin perustuva määritelmä HITS-algoritmin autkoriteetti- ja hubiarvoihin. Vastaavasti kuten HITS, myös SALSA on hakukohtainen, ja alkaa ympäristögraafin $\mathcal{N}$ muodostamisella hakutuloksista. Tämän jälkeen ympäristögraafi $\mathcal{N}$ jaetaan kolmeen osaan: hubisolmujen joukkoon $\mathcal{V}_h$, joka sisältää solmut joiden lähtöaste $d_O(n_i) > 0$, auktoriteettisolmujen joukkoon $\mathcal{V}_a$, joka sisältää solmut joiden tuloaste $d_I(n_i) > 0$, sekä nuolien joukkoon $\mathcal{E}$. Näistä muodostetaan suuntaamaton kaksimoodinen graafi $\mathcal{H}$, joka koostuu joukkojen $\mathcal{V}_h$ ja $\mathcal{V}_a$ solmuista uudelleen järjestettynä, sekä suuntaamattomat viivat, jotka määritellään seuraavasti

Määritelmä 3.3.1. Jos ympäristögraafissa $\mathcal{N}$ on nuoli somusta n_i solmuun n_j, niin graafissa $\mathcal{H}$ on viiva solmusta $n_i^{(h)}$ solmuun $n_j^{(h)}$, missä $n_i^{(h)}$ vastaa solmua n_i joukossa $\mathcal{V}_h$, ja $n_j^{(a)}$ solmua n_j joukossa $\mathcal{V}_a$.

Näin ollen graafissa on viivoja ainoastaan joukkojen $\mathcal{V}_h$ ja $\mathcal{V}_a$ välillä, mutta ei lainkaan kummankaan joukon sisällä. Koska ympäristögraafin $\mathcal{N}$ solmuilla voi olla sekä positiivinen tuloaste että lähtöaste, toteuttaa solmujen lukumäärä g_H graafissa $\mathcal{H}$ $g_H \leq 2 * g_N$, missä g_N on solmujen lukumäärä ympäristögraafissa $\mathcal{N}$. Kun graafissa $\mathcal{H}$ liikkuu satunnaisesti viivojen suunnassa, siirtyy joka toisella askeleella solmuun joukossa $\mathcal{V}_a$, ja joka toisella askeleella solmuun joukossa $\mathcal{V}_h$. Kun graafissa edetään satunnaisesti aina kaksi askelta kerrallaan, syntyy lähtösolmusta riippuen kaksi satunnaiskulkua, joita kuvaavat Markovin ketjut määrittävät auktoriteetti- ja hubipisteet. Kun tämä kirjoitetaan matriisimuodossa, päästään lopulta yhtälöihin

$$\begin{aligned} x^{(k)} &= L_{row}^T L_{col} x^{(k-1)} \\ x^{(k)} &= \frac{x^{(k)}}{||x^{(k)}||_1} \end{aligned} \tag{3.3.1}$$

$$\begin{aligned} y^{(k)} &= L_{col} L_{row}^T y^{(k-1)} \\ y^{(k)} &= \frac{y^{(k)}}{||y^{(k)}||_1} \end{aligned} \tag{3.3.2}$$

Nämä yhtälöt ovat tismalleen samat kuin satunnaistetun HITS-algoritmin yhtälöt 3.2.14 ja 3.2.15 parametrilla $\xi = 1$. Näin ollen kahdella eri muokkauksella on päädytty käytännössä samaan algoritmiin. SALSA-algoritmin ongelmana on auktoriteetti- ja hubimatriisien mahdollinen redusoituvuus, koska satunnaistetun HITS-algoritmin tapaan kaikkiin solmuihin arvostusta jakavaa muokkausta tehdä. Alunperin tämä ratkaistiin jakamalla graafi $\mathcal{H}$ yhtenäisiin osagraafeihin, joille jokaiselle laskettiin arvostukset erikseen, ja lopuksi kaikki nämä yhdistettiin. Laskennallisesti tämä on kevyempää, koska keralla käsiteltävät graafit ovat pienempiä. Toisaalta kuitenkin graafin hajottaminen yhtenäisiin osagraafeihin vaatii jonkin verran laskentaa, jolloin saavutettavat hyödyt vähenevät tai katoavat kokonaan, riippuen graafin $\mathcal{N}$ rakenteesta. Tässä työssä tyydytään käsittelemään laskentaa satunnaistetulla HITS-algoritmilla, jonka avulla saadaan vastaavia tuloksia ilman graafin hajottamista.

4. DATAN KUVAUS

Tarkasteltavana datana käytetään Tampereen teknillisen yliopiston kurssitarjonnan esitietoketjuja. Esitietojen periaate on, että kurssin suorittamisen edellytyksenä on tietty määrä suoritettuja aiempia kursseja, joista saadut tiedot ovat pohjana seuraavalla kurssilla käsiteltäville asioille. Näin esitietoketju muodostaa graafin, jossa kurssit ovat solmuja ja esitietovaatimukset solmujen välisiä nuolia. Yhteensä käytettävässä datassa on 914 kurssia, joille on yhteensä 1502 esitietoa. Näin ollen esitiedoista syntyvän graafin tiheydeksi saadaan

$$\Delta = \frac{1502}{914 * 913} \approx 0,0018 \tag{4.0.1}$$

Näin ollen syntyvä graafi on erittäin harva, sillä vain 0.18% mahdollisista nuolista on olemassa. Tämä selittyy osaltaan esitietoketjujen luonteella, koska näistä muodostuvassa graafissa ei voi olla solmujen välillä nuolta molempiin suuntiin. Myöskin ketjut ovat hierarkisia, eikä takaisinlinkitystä tapahdu ollenkaan. Toinen tekijä on datan epätäydellisyys, koska tämä ei kata kaikkia esitietoja. Esimerkiksi matematiikan ja fysiikan peruskurssit ilmaistaan esitiedoissa kokonaisuuksina, joten yksittäiset kurssit eivät näy edistyneempien kurssien esitiedoissa. Graafin harvuus ei kuitenkaan tuota ongelmia arvostusalgoritmien soveltamiselle, koska nämä ovat alunperin suunniteltu internetin verkostojen tutkimiseen, jotka ovat myös erittäin harvoja graafeja. On arvioitu, että nettisivuilla on keskimäärin noin kymmenen linkkiä, joten internetin sisältäessä miljardeja eri sivuja, on syntyvä graafi todella harva.

Määritelmä 4.0.2. Esitietoketjuista muodostuu graafi $\mathcal{G} = \{\mathcal{V}, \mathcal{L}\}$, missä solmujen joukko

$$\mathcal{V} = \{n_1, n_2, \ldots, n_g\}$$

on kurssit järjestettynä kurssitunnuksen mukaisesti aakkosjärjestykseen. Nuolien joukko $\mathcal{L}$ määritellään

$$\mathcal{N} = \{< n_i, n_j > \mid \text{kurssi } n_i \text{ on kurssin } n_j \text{ esitieto}\}$$

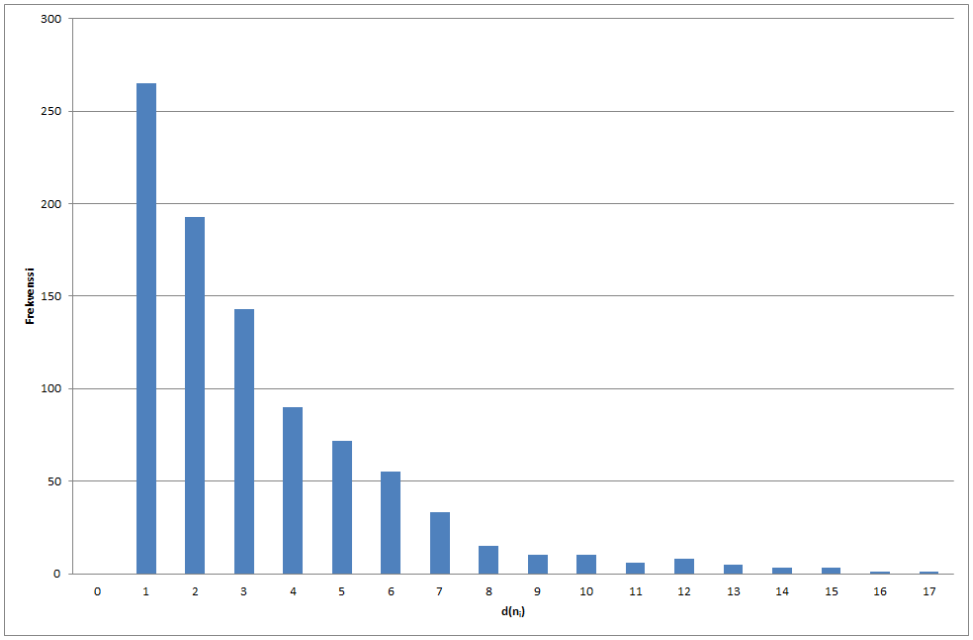
Tästä graafista muodostuu vieruspistematriisi, joka on seuraavanlainen.

Määritelmä 4.0.3. Esitietoketjuista muodostuvan graafin vieruspistematriisin L alkiot l_{ij} määritellään

$$l_{ij} = \begin{cases} 1 & \text{jos kurssi i on kurssin j esitieto} \\ 0 & \text{muulloin} \end{cases}$$

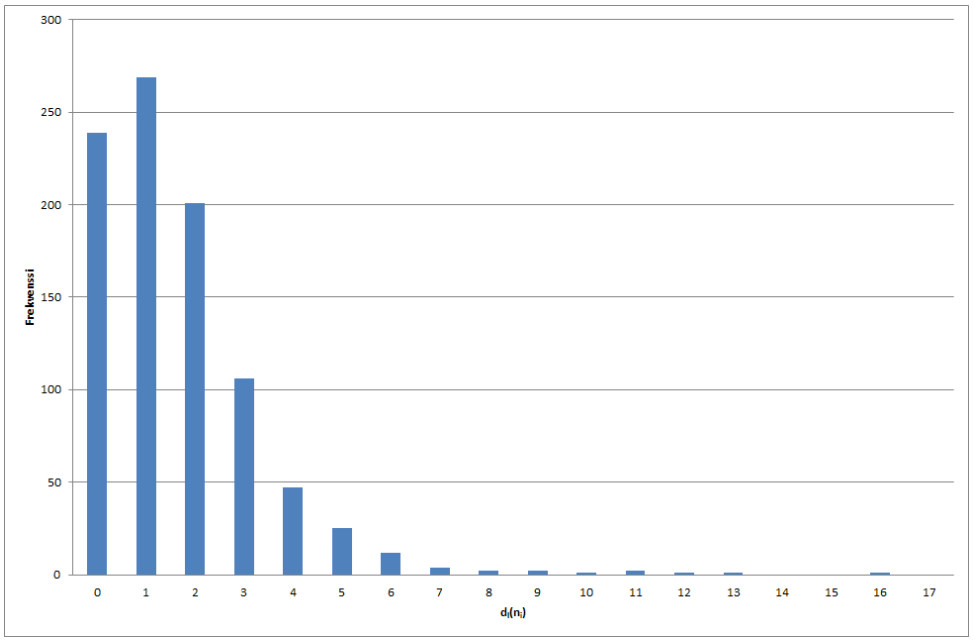
Esitietokurssit ovat luonteeltaan joko pakollisia tai suositeltavia. Arvostuksia laskiessa ei tehdä eroa onko esitieto pakollinen vai suositeltava, vaan solmujen välinen nuoli on olemassa molemmissa tapauksissa. Myöskin kurssit joilla ei ole esitietoja tai jotka eivät ole minkään kurssin esitietona jätetään pois. Nämä saisivat triviaalisti kaikki saman arvon, ja vaikuttaisivat muiden kurssien arvoihin ainoastaan skaalaavasti. Myöskin kurssi KIE-6201 jätetään pois, koska tällä on merkattuna ainoastaan itsensä esitiedoksi, ja tämä vääristää tuloksia.

Tarkasteltaessa muodostuvan graafin tunnuslukuja, saadaan yleiskäsitys datan sisällöstä. Graafin solmujen asteiden jakauma on esitetty kuvassa 4.1. Koska datassa on huomioitu ainoastaan kurssit joilla on esitietoja, tai jotka ovat jollekin toiselle kurssille esitietona, ei graafissa ole solmuja joiden aste olisi nolla. Solmun asteiden jakauma muistuttaa paljolti eksponenttijakaumaa, ja valtaosa solmuista on sellaisia, joihin liittyy ainoastaan muutamia nuolia.



Kuva 4.1: Esitietoketjuista muodostuvan graafin solmujen asteen jakauma

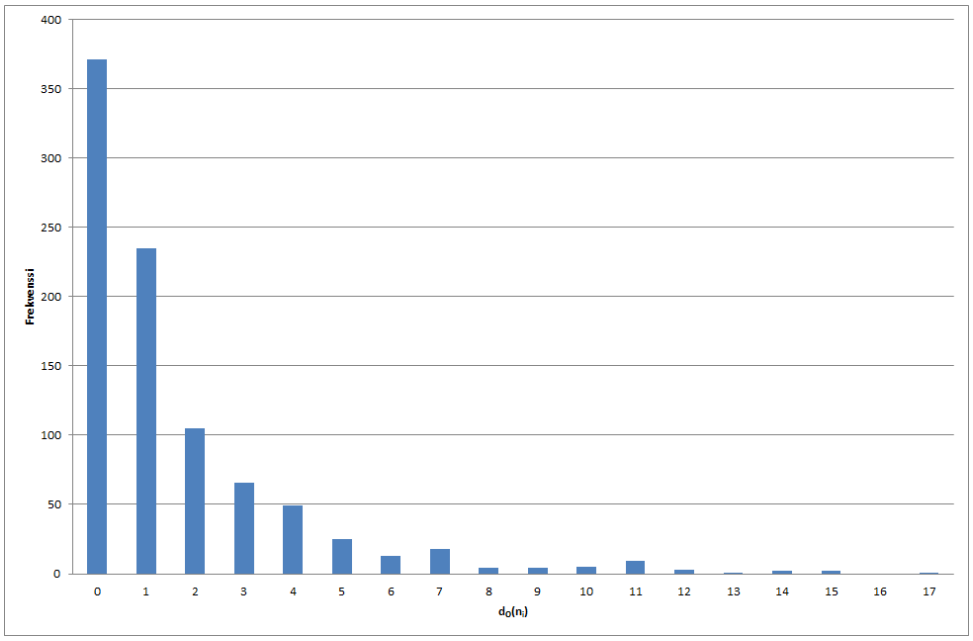
Koska kyseessä on suunnattu graafi, on solmujen astetta mielenkiintoisempaa tarkastella solmujen tulo- ja lähtöasteita. Näin määritellylle verkostolle tuloaste tarkoittaa kurssiin liittyvien esitietojen lukumäärää, ja lähtöaste niiden kurssien lukumäärää, joille kyseinen kurssi on esitietona. Tulo- ja lähtöasteiden jakaumat on esitetty kuvissa 4.2 ja 4.3. Näistä havaitaan, että huomattavasti useamman solmun lähtöaste on nolla, kuin tuloaste on nolla. Tämä tarkoittaa siis että graafissa on enemmän nieluja kuin lähteitä. Tämä on luonnollisesti varsin odotettua, koska pienempi joukko peruskursseja toimii monien edistyneempien kurssien lähtötietoina, ja nämä ketjut päättyvät aina johonkin yksittäiseen kurssiin. Muuten sekä tulo- että lähtöasteiden jakaumat muistuttavat solmujen asteiden tapaan hyvin pitkälti eksponenttijakaumaa, joskin lähtöasteiden jakauma painottuu enemmän suurille arvoille. Tämä tarkoittaa siis sitä, että datassa on enemmän kursseja, jotka toimivat monien kurssien esitietona, kuin kursseja joilla on paljon esitietoja.



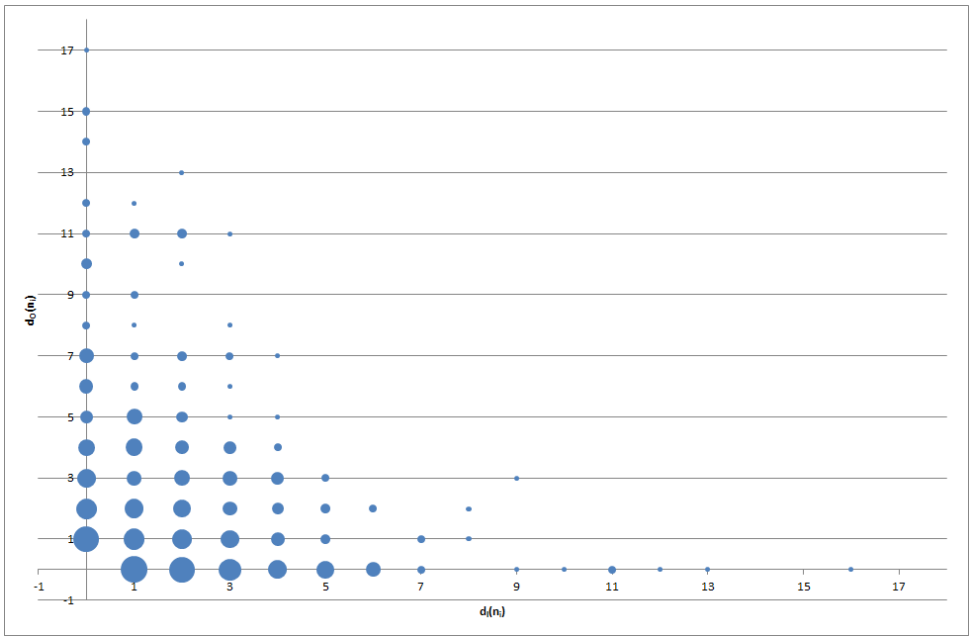
Kuva 4.2: Esitietoketjuista muodostuvan graafin solmujen tuloasteen jakauma

Tarkasteltaessa solmujen tulo- ja lähtöasteiden yhteisjakaumaa nähdään vielä selkeämmin jakauman painottuminen pienille asteluvuille. Huomion arvoista on myös se, että tulo-ja lähtöasteiden välinen riippuvuus on jokseenkin käänteinen, eikä datassa ole juurikaan solmuja joilla olisi sekä korkea lähtöaste, kuin korkea tuloaste. Tämä siis tarkoittaa että ei ole juurikaan kursseja, joilla olisi paljon esitietoja, ja jotka toimisivat vielä useiden kurssien esitietoina. Tämä yhteisjakauma on esitetty kuvassa 4.4, ja tässä kuvassa ympyrän halkaisija kuvaa kyseisen tulo- ja lähtöastekombinaation frekvenssiä datassa.

Laskettaessa graafin modulaarisuutta Q Gephin käyttämällä algoritmilla, saadaan



Kuva 4.3: Esitietoketjuista muodostuvan graafin solmujen lähtöasteen jakauma



Kuva 4.4: Esitietoketjuista muodostuvan graafin solmujen tulo- sekä lähtöasteen yhteisjakauma

peräti 64 eri klusteria, ja graafin modulaarisuudeksi tulee

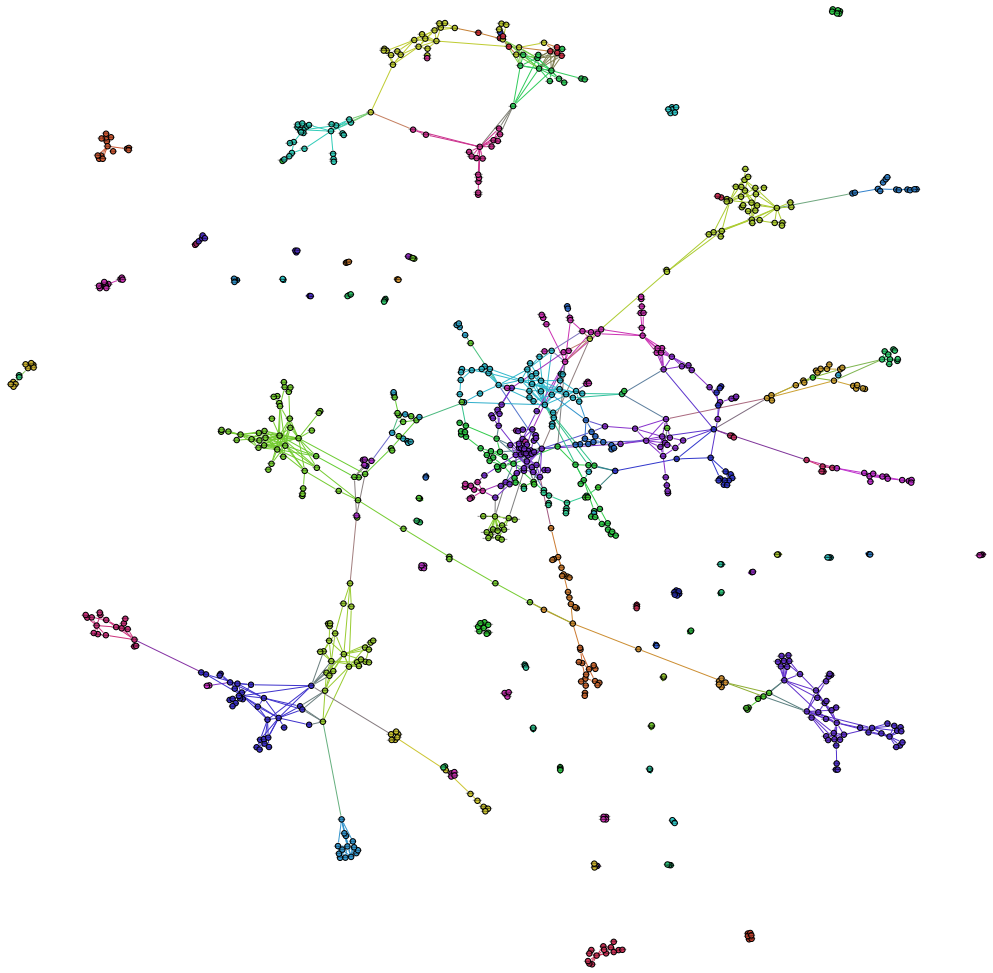
$$Q = 0.908 \tag{4.0.2}$$

Tämä tarkoittaa siis sitä, että graafi koostuu lukuisista erillisistä joukoista, jotka ovat melko harvoin yhteydessä toisiinsa. Kursseja tarjoavia laitoksia on yhteensä 45 kappaletta, joten on selvää että laitosten sisällä esitietoketjut eivät rakennu hierarkisesti. Tämä on kuitenkin varsin luonnollista, sillä eri laitokset tarjoavat varsin erilaisia kokonaisuuksia. Muodostuvan graafin halkaisija $d(\mathcal{G})$, eli pisimmän esitietoketjun pituus on

$$d(\mathcal{G}) = 6 \tag{4.0.3}$$

Tämä vahvistaa kuvaa erillisistä joukoista koostuvasta graafista, sillä pisimmänkin esitietoketjun pituus on varsin pieni.

Piirrettäessä Gephin ForceAtlas2-algoritmillä kuva käsiteltävästä verkostosta, nähdään visuaalisesti mitä tämä tarkoittaa. Tämä on esitettynä kuvassa 4.5, jossa eri laitosten tarjoamat kurssit ovat esitettynä eri väreillä. Eri opintokokonaisuudet muodostavat enemmän tai vähemmän tiiviitä ryppäitä, jotka ovat melko vähän yhteydessä toisiin kokonaisuuksiin. Näiden suurempien kokonaisuuksien lisäksi on useita pienempiä kokonaisuuksia, jotka ovat täysin irrallaan muista. Suurempana irrallisena joukkona näkyy Porin yksikön opintokokonaisuudet, jotka eivät ole missään yhteydessä muuhun yliopiston tarjontaan. Nämä näkyvät neljän suuremman kokonaisuuden yhdistelmänä kuvan yläreunassa.



Kuva 4.5: Esitietoketjuista muodostuva graafi

5. TULOKSET

Tulokset laskettiin Pagerankilla ja Cheirankilla käyttäen parametreina $\alpha_1 = 0,85$, $\alpha_2 = 0,95$ ja $\alpha_3 = 0,99$. HITS-algoritmista käytettiin tämän modifioitua sekä satunnaistettua variaatiota, käyttäen parametreina $\xi_1 = 0,85$, $\xi_2 = 0,95$ ja $\xi_3 = 0,99$ molemmille variaatioille. Tuloksia tarkasteltaessa pureuduttiin eri kurssien arvojärjestykseen eri algoritmeilla, sekä näiden tulosten tulkintaan. Lisäksi tarkasteltiin auktoriteetti- ja hubipisteiden, kuin myös Pagerankin ja Cheirankin välistä yhteyttä, sekä eri algoritmien laskentanopeuksia ja iteraatiolukumääriä. Eri parametreilla ei arvostusjärjestys juurikaan vaihdellut, ja Pagerankin ja Cheirankin tapauksessa suuruusluokatkin pysyivät melko samoina. HITS-algoritmin yhteydessä havaittiin, että algoritmin kaksi eri variaatiota antavat hyvinkin erilaisia tuloksia, ja todettiin modifioifun HITS-algoritmin olevan melko herkkä manipuloinnille. Satunnaistetulla HITS-algoritmilla saatiin hyvinkin saman suuntaisia tuloksia kuin Pagerankilla ja Cheirankilla, mikä on melko luonnollista ottaen huomioon kuinka paljon nämä algoritmit muistuttavat toisiaan. Tulkinnan helpottamiseksi tulokset esitetään kymmenkantaisena logaritmina. Arvostusten lisäksi tuloksissa on listattuna kurssiin liittyvien esitietojen lukumäärä, sekä niiden kurssien lukumäärä, joille kyseinen kurssi on esitietona. Näin nähdään lisätietoa siitä, miten arvostusalgoritmit toimivat.

5.1 Iteraatiot ja laskenta-ajat

Käytetyt algoritmit toteutettiin MATLAB R2010a-ohjelmistolla, ja laskenta suoritettiin muutaman vuoden ikäisellä tietokoneella. Alkuarvauksena kertomenetelmälle käytettiin jokaisen algoritmin yhteydessä tasajakautunutta vektoria $\frac{1}{n}e^T$. Suppenemiskriteerinä käytettiin $\epsilon = 10^{-10}$, eli iterointi lopetettiin kun $||x^T_{(k+1)} - x^T_k||_1 < \epsilon$. Iteraatiolukumäärät sekä laskenta-ajat eri algoritmeille ja eri parametrien α ja ξ arvoille ovat esitettynä taulukoissa 5.1, 5.2 ja 5.3

Pagerankin ja Cheirankin tapauksessa nähdään selkeästi parametrin α vaikutus kertomenetelmän suppenemiseen. Kuten aiemmin todettiin, suuremmilla parametrin α arvoilla tarvitaan enemmän iteraatioita halutun tarkkuuden saavuttamiseksi, joten luonnollisesti laskenta-aikakin kasvaa. Tässä tapauksessa laskenta-aikojen ero eri parametrin α arvoilla on varsin pieni, ja ylipäätään laskenta tapahtuu huomattavan nopeasti. Cheirankin tapauksessa iteraatio

Taulukko 5.1: Iteraatiolukumäärät ja laskenta-ajat eri algoritmeilla parametreilla $\alpha = \xi = 0,85$

	Iteraatioita (kpl)	Laskenta-aika (s)
Pagerank	18	0,0444
Cheirank	23	0,0551
Modifioitu HITS auktoriteetti	45	10,7446
Modifioitu HITS hubi	46	10,9646
Satunnaistettu HITS auktoriteetti	7	0,2488
Satunnaistettu HITS hubi	6	0,2120

Taulukko 5.2: Iteraatiolukumäärät ja laskenta-ajat eri algoritmeilla parametreilla $\alpha = \xi = 0,95$

	Iteraatioita (kpl)	Laskenta-aika (s)
Pagerank	20	0,0470
Cheirank	26	0,0611
Modifioitu HITS auktoriteetti	45	10,7296
Modifioitu HITS hubi	46	11,3453
Satunnaistettu HITS auktoriteetti	10	0,3499
Satunnaistettu HITS hubi	8	0,2813

Taulukko 5.3: Iteraatiolukumäärät ja laskenta-ajat eri algoritmeilla parametreilla $\alpha = \xi = 0,99$

	Iteraatioita (kpl)	Laskenta-aika (s)
Pagerank	21	0,0482
Cheirank	28	0,0634
Modifioitu HITS auktoriteetti	45	10,5439
Modifioitu HITS hubi	46	10,9652
Satunnaistettu HITS auktoriteetti	26	0,9228
Satunnaistettu HITS hubi	17	0,6212

suppenee hieman hitaammin, joka selittyy tulo- ja lähtöasteiden erilaisilla jakaumilla. On kuitenkin syytä muistaa, että Pagerankia ja Cheirankia ei sinällään tule tarkastella erillisinä, sillä kyseessä on kuitenkin sama algoritmi eri suuntaisille verkostoille. Tässä tapauksessa verkon suunta oli vapaasti valittavissa, joten näiden kahden ero on täysin terminologinen ja tarpeen vain kuvastamaan näiden kahden konseptillista eroa.

Tarkasteltaessa iteraatiolukumääriä ja laskenta-aikoja HITS-algoritminn tapauksessa, huomataan että käytettävien variaatioiden välillä on dramaattiset erot. Modifioitu HITS vaatii huomattavasti enemmän iteraatioita, ja moninkertaisen laskenta-ajan verrattuna satunnaistettuun HITS-algoritmiin. Modifioidun variaation tapauksessa parametri ξ ei vaikuta lainkaan tarvittavien iteraatioiden lukumäärään, ja laskenta-aikojen erot ovat käytännössä mitättömiä. Tämä indikoi ongelmaa tämän variaation soveltamisessa, ja tuloksia tarkasteltaessa huomataankin että nämä käyttäytyvät varsin omituisesti. Iteraatiolukumäärää radikaalimmin noussut laskenta-aika johtuu osaltaan siitä, että modifioidun HITS-algoritmin tapauksessa kullakin iteraatiolla joudutaan operoimaan matriiseilla, joiden kaikki alkiot ovat erisuuria kuin nolla. Tämä osaltaan pidentää tarvittavaa laskenta-aikaa, tehden tästä monikymmenkertaisen muihin algoritmeihin nähden.

Satunnaistetun HITS-algoritmin tapauksessa kertomenetelmä suppenee jo varsin pienellä iteraatiolukumäärällä, ja laskenta-aika on murto-osa modifioidun variaation vastaavasta. Parametrin ξ vaikutus on kaikkein radikaalein satunnaistetun HITS-algoritmin tapauksessa, ja parametrin ξ kasvaessa iteraatioita tarvitaan moninkertainen määrä. Vaikka satunnaistettu HITS muistuttaakin melko läheisesti Pagerankia, kuluu sen laskemiseen yli kymmenkertainen aika samalla iteraatiolukumäärällä. Tässä nousee esiin Pagerankin laskennallinen etu, kun laskenta voidaan hajoittaa yhtälön 3.1.6 mukaisesti harvan matriisin ja vektorien kerto-ja yhteenlaskuksi. Satunnaistetun HITS-algoritmin tapauksessa operoidaan myös harvoilla matriiseilla, mutta jokaisella iteraatiolla tarvitsee kuitenkin suorittaa matriisi-matriisi kertolaskuja, joiden laskeminen vie moninkertaisen ajan verrattuna Pagerankin matriisi-vektori kertolaskuun. On selvää, että tämä tulee ongelmaksi tarkasteltaessa suurempia verkostoja, mikä on osaltaan syy siihen miksi HITS-algoritmi on alunperinkin määritelty hakukohtaiseksi, jolloin käsiteltävä graafi ei kasva liian suureksi.

5.2 Pagerank

Kurssien Pagerank-arvot parametrilla, $\alpha = 0,85$, $\alpha = 0,95$ ja $\alpha = 0,99$ ovat esitettynä taulukossa 5.4. Tulokset on järjestetty parametrilla $\alpha = 0,99$ saatavien tulosten mukaiseen suuruusjärjestykseen.

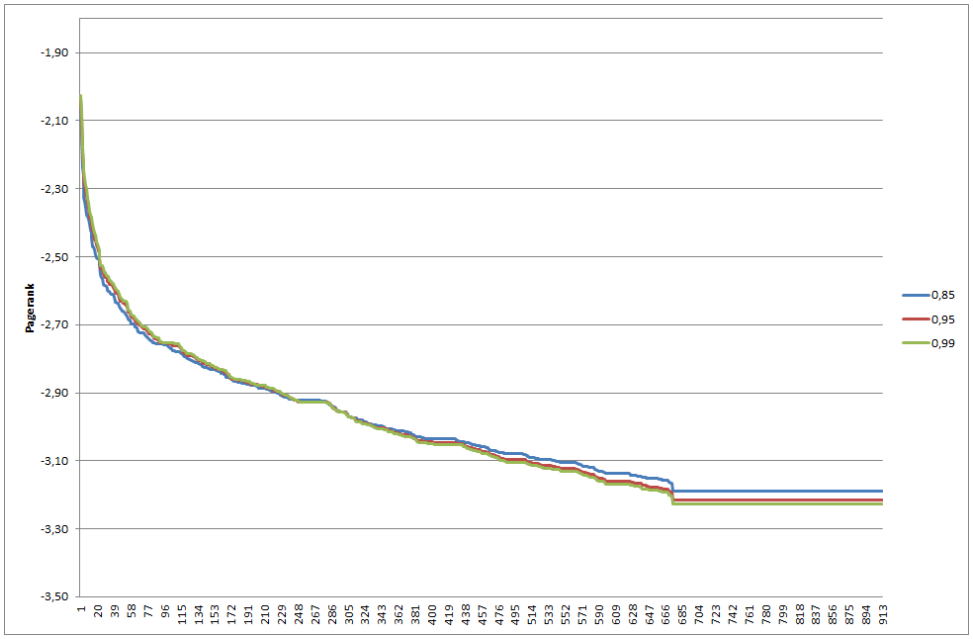
Taulukko 5.4: Kurssien 15 suurinta Pagerank-arvoa parametreilla $\alpha = 0,85$, $\alpha = 0,95$ ja $\alpha = 0,99$

Pagerank (log_{10})					
Kurssi	$\alpha = 0,85$	$\alpha = 0,95$	$\alpha = 0,99$	Esitietoja	Esitietona
BIO-4700	-2,07	-2,04	-2,03	16	0
PAK-2070	-2,13	-2,10	-2,08	13	0
RTEK-3880	-2,24	-2,20	-2,18	12	0
OHJP-3600	-2,26	-2,24	-2,23	11	0
ELEP-1850	-2,33	-2,27	-2,25	3	0
ELE-6286	-2,36	-2,29	-2,27	4	0
BME-4606	-2,34	-2,30	-2,29	5	0
ENER-3051	-2,38	-2,32	-2,30	2	0
OHJ-1500	-2,38	-2,34	-2,33	9	0
KEM-4050	-2,39	-2,36	-2,34	6	0
ELEP-4810	-2,43	-2,38	-2,37	4	0
ENER-3010	-2,42	-2,39	-2,38	8	1
TEL-1480	-2,47	-2,41	-2,38	3	0
SMG-4550	-2,40	-2,39	-2,39	6	0
ELE-6216	-2,47	-2,42	-2,41	4	0

Näistä havaitaan, että erityisesti paljon esitietoja sisältävät syventävät kurssit nousevat kärkeen Pagerankissa. Nämä ovat yhtä poikkeusta lukuunottamatta kaikki nieluja, joten nämä voidaan tulkita olevan viimeisiä kursseja opintokokonaisuuksista mitä opiskelijat suorittavat. Pagerankin vahvuutena tässä nousee esille, että vaikkakin arvostetuimmilla kursseilla on suurimmat määrät esitietoja, mukaan nousee myöskin kursseja joilla on vain muutamia esitietoa. Tämä korostaa arvostettujen esitietojen merkitystä, ja alleviivaa Pagerankin alkuperäisen määritelmän ajatusta. Huomionarvoista on parametrin α vaikutus: pienemmillä α:n arvoilla korostuvat hieman kurssit joilla on enemmän esitietoja, ja suuremmilla arvoilla ne joilla on arvostetumpia esitietoja. Tämä on myöskin hyvin linjassa määritelmien kanssa, koska parametrilla α nimenomaan säädellään linkkirakenteen ja satunnaissiirtymän välistä painoarvoa. Muutokset eri kurssien arvostuksessa eri parametrin α arvoilla ovat kuitenkin melko pieniä, ja suuruusluokkakin vaihtelee vain nimellisesti. Pagerank-arvojen jakauma kullakin parametrin α arvolla on esitetty kuvassa 5.1.

Tästä nähdään että jakauma ei muutu nimeksikään eri parametrin α arvoilla, eikä myöskään kokoluokka. Yksittäisten kurssien välillä järjestys kuitenkin muuttuu. Kurssit joilla ei ole esitietoja, ovat graafissa lähteitä, ja nämä kaikki saavat saman Pagerank-arvon.

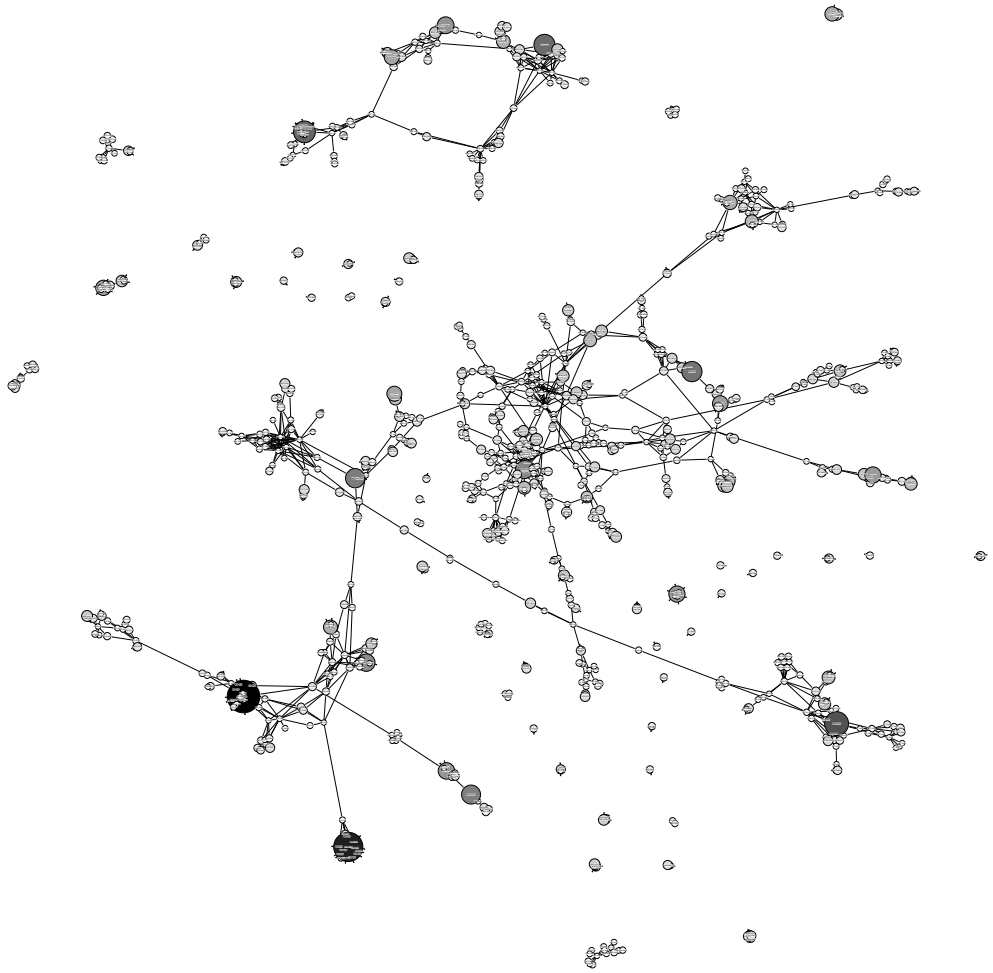
Pagerank-arvostuksen visualisointi on esitetty kuvassa 5.2. Tässä solmun koko ja



Kuva 5.1: Pagerank-arvojen jakauma kullakin parametrin α arvolla

väri kuvaavat sen Pagerank-arvoa, ja mitä korkeampi on kyseisen solmun arvostus, sitä suurempi ja väriltään lähempänä mustaa se on graafissa.

Tässä sovelluskohteessa parametrin α tulkinta tällaisenaan on melko hankalaa, sillä yksittäisen opiskelijan etenemistä verkostossa tarkasteltaessa ei ole loogista tutkia puhtaasti linkkirakennetta, mutta toisaalta täysin satunnainen siirtyminen on myös ajatuksena melko huono. Tämä osaltaan siksi, että verkosto ei kata kaikkia peruskursseja, ja toisaalta siksi, että opiskelijan etenemistä tässä verkostossa ohjaa kuitenkin enemmän ulkopuoliset tutkintovaatimukset. Pelkästään verkoston ominaisuuksien valossa tutkittaessa on syytä ottaa tarkastelussa mukaan myös eri kurssien Cheirank-arvot, joiden avulla saadaan monipuolisempi käsitys näiden roolista muodostuvassa verkostossa. Tällöin myös päästään eroon ongelmista tulkittaessa verkon suunnan määrittelyä, sillä Cheirank määritelmänsä mukaisesti vastaa Pagerank-algoritmia verkostolle, jonka suunta on käännetty. Näin ollen tässä tapauksessa nuolien suunnan määrittelyllä ei ole merkitystä, ainoastaan tulkinnassa on huomioitava kumpaa arvostusta käsitellään tuloksia tulkittaessa.



Kuva 5.2: Kuva graafista, kun solmun koko ja väri kuvastaa kyseisen solmun Pagerank-arvoa

5.3 Cheirank

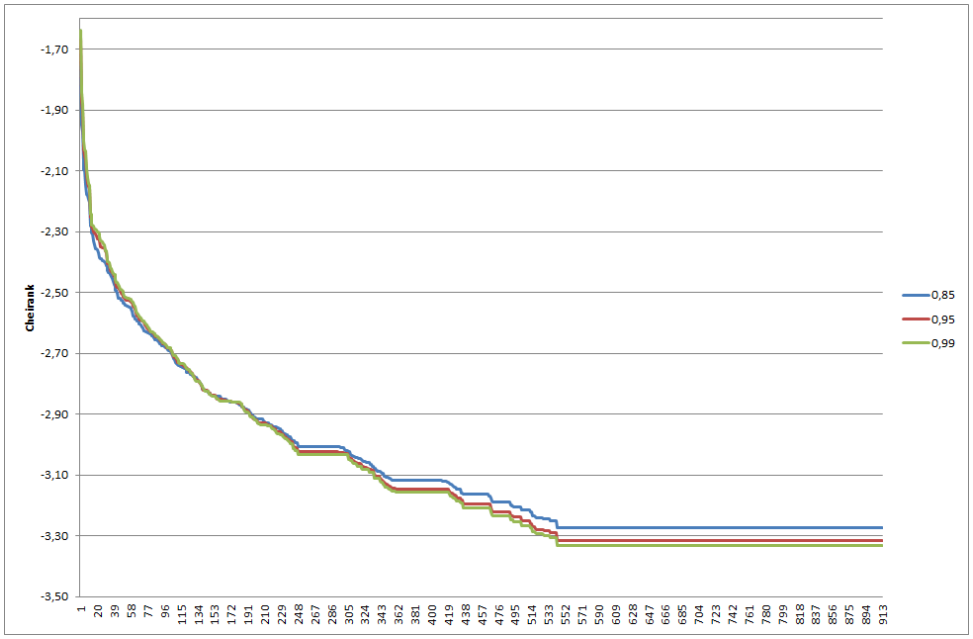
Kurssien Cheirank-arvot parametrilla, $\alpha = 0,85$, $\alpha = 0,95$ ja $\alpha = 0,99$ ovat esitettynä taulukossa 5.4. Tulokset on järjestetty parametrilla $\alpha = 0,99$ saatavien tulosten mukaiseen suuruusjärjestykseen.

Taulukko 5.5: Kurssien 15 suurinta Cheirank-arvoa parametreilla $\alpha = 0,85$, $\alpha = 0,95$ ja $\alpha = 0,99$

Cheirank (log_{10})					
Kurssi	$\alpha = 0,85$	$\alpha = 0,95$	$\alpha = 0,99$	Esitietoja	Esitietona
OHJ-1010	-1,76	-1,67	-1,64	0	7
TME-1100	-1,95	-1,87	-1,84	0	4
OHJ-1100	-1,97	-1,91	-1,88	1	11
SMG-1100	-2,07	-2,01	-1,99	0	7
MATP-1311	-2,10	-2,03	-2,00	0	12
KEM-1410	-2,15	-2,06	-2,03	0	6
SGN-1201	-2,10	-2,05	-2,04	0	17
MPR-4010	-2,18	-2,12	-2,09	0	7
TME-1301	-2,19	-2,14	-2,12	1	5
TETA-1010	-2,18	-2,15	-2,14	0	12
OHJ-1150	-2,20	-2,16	-2,15	1	11
PAK-1010	-2,28	-2,25	-2,24	0	10
RTEK-2011	-2,28	-2,25	-2,24	2	11
MATHM-37100	-2,31	-2,28	-2,28	2	11
MAT-10412	-2,36	-2,30	-2,28	0	6

Kuten loogisesti voidaan odottaakin, nousee Cheirank-arvoissa korkeimmalle peruskurssit, jotka toimivat monille kursseille esitietoina. Nämä solmut ovat enimmäkseen lähteitä, ja toimivat esitietoketjujen alkupisteinä. Vastaavasti kuten Pagerankilla, huomataan että listalle nousee myös kursseja joilla on melko vähän esitietoja, mutta jotka ovat esitietona arvostetuille kursseille. Parametrin α vaikutus on luonnollisesti vastaava kuten aiemminkin, ja suuremmilla parametrin α arvoilla korostuvat ne kurssit, jotka ovat arvostetummille kursseille esitietona. Cheirank-arvojen jakauma on esitetty kuvassa 5.3.

Tämä jakauma muistuttaa myös hyvin paljon eksponenttijakaumaa, mutta Pagerankiin verrattuna arvostetuimmat kurssit saavat hieman korkeampia arvoja, ja vastaavasti vähiten arvostetut kurssit hieman matalampia. Tämä johtuu eroista solmujen tulo- ja lähtöasteiden jakaumassa, ja lähtöjakauma painottuu enemmän suurille arvoille. Kurssit jotka eivät ole minkään kurssit esitietona, ovat graafissa nieluja, ja nämä saavat kaikki saman Cheirank-arvon. Koska graafissa on enemmän nieluja kuin lähteitä, on näiden saman minimiarvostuksen saavien kurssien osuus



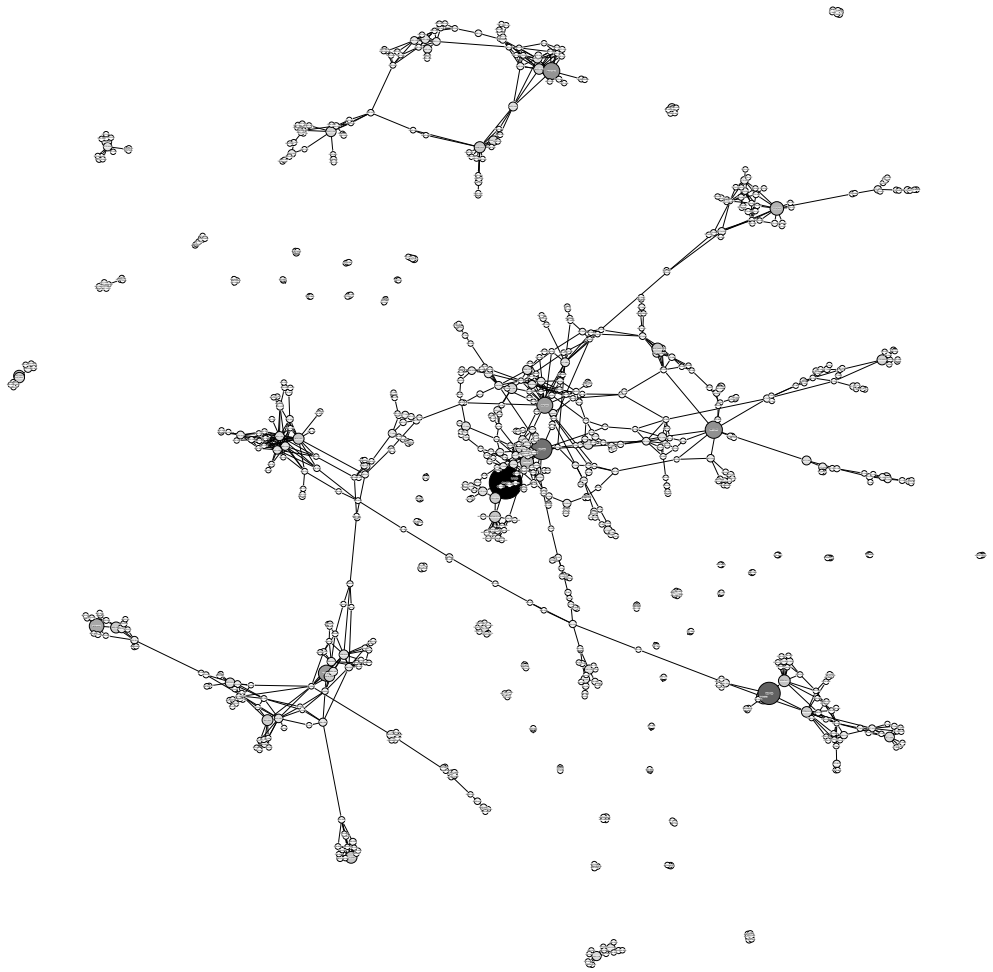
Kuva 5.3: Cheirank-arvojen jakauma kullakin parametrin α arvolla

jakaumasta luonnollisesti suurempi Cheirankin tapauksessa kuin Pagerankin. Mielenkiintoinen ilmiö ovat myös jakaumassa esiintyvät muutamat portaat, joissa on kymmeniä kursseja lähes samalla arvostuksella. Tämä sama ilmiö toistuu kaikilla parametrin α arvoilla.

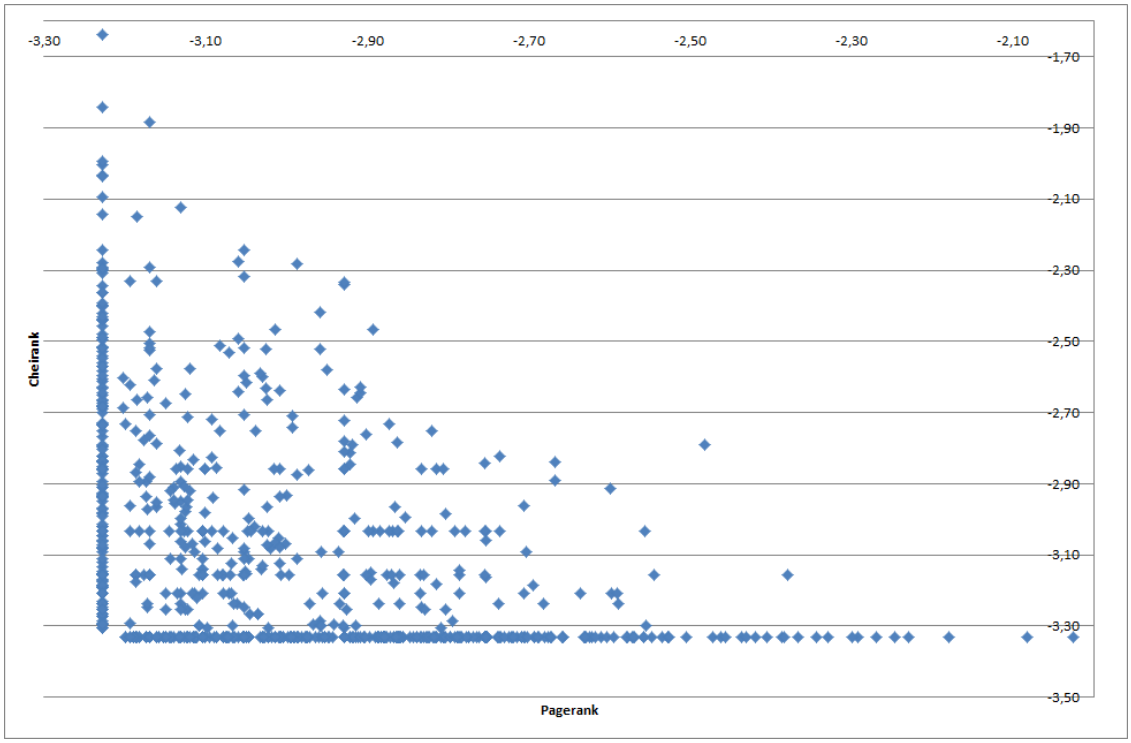
Cheirank-arvostuksen visualisointi on esitetty kuvassa 5.4. Tässä solmun koko ja väri kuvaavat sen Cheirank-arvoa, ja mitä korkeampi on kyseisen solmun arvostus, sitä suurempi ja väriltään lähempänä mustaa se on graafissa.

Kurssien Pagerank- ja Cheirank-arvojen välinen riippuvuus on esitettynä kuvassa 5.5. Tämä jakauma on lähes identtinen kaikilla parametrin α arvoilla, joten on mielekästä tarkastella ainoastaan yhtä tapausta. Tässä sovelluskohteessa on loogista tarkastella enemmän graafin linkkirakenteen mukaan tulevaa arvostusta, joten tarkasteltavaksi skenaarioksi on valittu tapaus $\alpha = 0,99$.

Kuvasta nähdään, että molemmilla akseleilla ääriarvot saavutetaan toisen akselin minimiarvossa, ja näiden kahden arvostuksen välinen riippuvuus vaikuttaa jossain määrin käänteiseltä. Tämä on luonnollista kun tarkastelee määrittelyjä, koska Pagerankin tapauksessa arvostetuimmat solmut olivat nieluja, ja Cheirankin tapauksessa lähteitä. Hajontaa on keskivaiheilla melko paljon, eli riippuvuus ei kuitenkaan ole puhtaasti käänteinen. Näin ollen voidaan todeta, että tarkastelemalla näitä kahta arvostusta yhdessä voidaan todella saada relevanttia lisätietoa graafin rakenteesta ja solmujen arvostuksesta verrattuna tilanteeseen, jossa molempia näitä arvostuksia tarkasteltaisiin erikseen.



Kuva 5.4: Kuva graafista, kun solmun koko ja väri kuvastaa kyseisen solmun Cheirank-arvoa



Kuva 5.5: Kurssien Pagerank- ja Cheirank arvojen yhteisjakauma parametrilla $\alpha = 0,99$

5.4 Modifioitu HITS

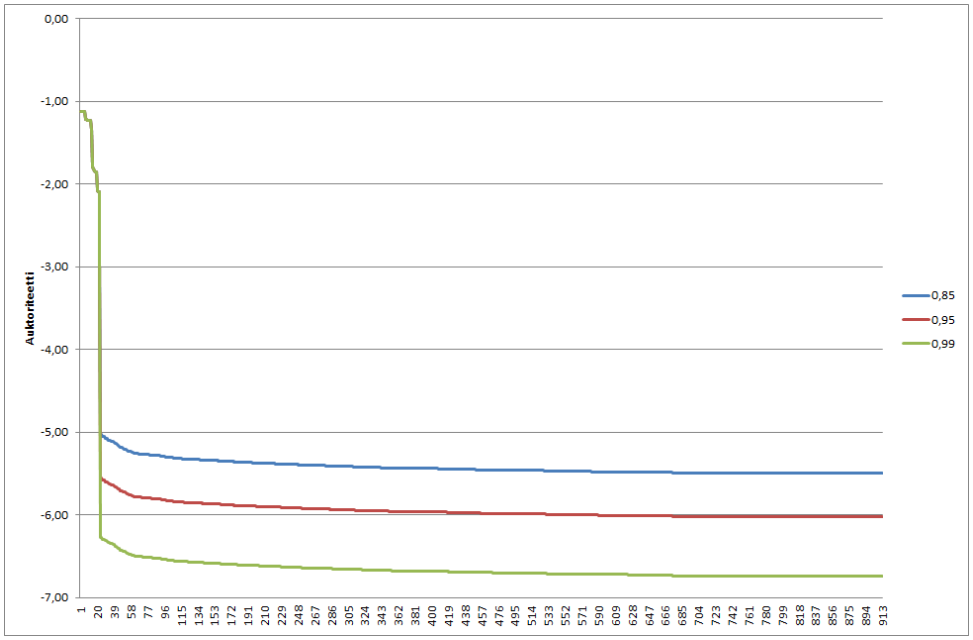
Kurssien auktoriteettiarvot parametrilla, $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$ ovat esitettynä taulukossa 5.6. Tulokset on järjestetty parametrilla $\xi = 0,99$ saatavien tulosten mukaiseen suuruusjärjestykseen.

Tarkasteltaessa suurimpia auktoriteettiarvoja modifioidulla HITS-algoritmilla, havaitaan että yhtä lukuunottamatta kaikki kurssit ovat saman laitoksen kursseja. Tarkasteltaessa lähtödataa havaitaan lisäksi, että myös kurssi BME-1120 liittyy Materiaaliopin laitoksen (MOL-xxxx) kursseihin, joten voidaan todeta että kaikki nämä liittyvät samaan kokonaisuuteen. Arvostus näyttää kasaantuvan tähän joukkoon, ja 15 suurimman auktoriteettiarvon summa on peräti $0,913$. Tämä tarkoittaa sitä, että peräti $91,3\%$ arvostuksesta kasaantuu näille 15 kurssille. Tarkasteltaessa tuloksia eteenpäin havaitaan, että 23 arvostetuinta kurssia ovat näitä MOL-xxxx ja BME-xxxx kursseja, jotka saavat peräti $99,7\%$ kaikesta arvostuksesta, ja kaikki loput saavat arvostuksia jotka ovat alle 10^{-5}. Näin ollen voidaan todeta että algoritmin soveltamisessa on ongelmia, sillä arvostuksen kasautuminen on varsin ei-toivottu ominaisuus. Auktoriteettiarvojen jakauma kullakin parametrin ξ arvolla on esitetty kuvassa 5.6.

Jakaumakuva näyttää hyvin arvostuksen kasautumisen. Muutamat kurssit saavat varsin korkeita arvoja, jonka jälkeen arvostus putoaa hyvin nopeasti kohti nollaa. Parametrilla ξ on myös varsin radikaali vaikutus jakaumaan. Suuremmilla

Taulukko 5.6: Kurssien 15 suurinta auktoriteettiarvoa modifioidulla HITS-algoritmilla parametreilla $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$

Auktoriteetti (log_{10})					
Kurssi	$\xi = 0,85$	$\xi = 0,95$	$\xi = 0,99$	Esitietoja	Esitietona
MOL-7106	-1,12	-1,12	-1,12	6	0
MOL-7207	-1,12	-1,12	-1,12	6	0
MOL-7307	-1,12	-1,12	-1,12	6	0
MOL-7506	-1,12	-1,12	-1,12	6	0
MOL-7606	-1,12	-1,12	-1,12	6	0
MOL-7706	-1,12	-1,12	-1,12	6	0
MOL-1600	-1,22	-1,22	-1,22	5	0
MOL-1530	-1,22	-1,22	-1,22	5	3
MOL-1500	-1,23	-1,23	-1,23	4	2
MOL-1520	-1,23	-1,23	-1,23	4	1
MOL-1510	-1,23	-1,23	-1,23	4	2
BME-1120	-1,23	-1,23	-1,23	4	1
MOL-5516	-1,35	-1,35	-1,35	3	1
MOL-1650	-1,35	-1,35	-1,35	3	0
MOL-3316	-1,80	-1,80	-1,80	2	0



Kuva 5.6: Auktoriteettiarvojen jakauma kullakin parametrin ξ arvolla modifioidulla HITS-algoritmilla

parametrin ξ arvoilla arvostukset putoavat suurimpien arvojen jälkeen hyvin nopeasti lähelle nollaa. Tämä kuvaa hyvin arvostuksen kasautumista ja polarisoitumista: vaikka pienimmille arvostuksille jäävä osuus putoaa merkittävästi suuremmilla parametrin ξ arvoilla, ei tämä erotu lainkaan suurimmissa arvostuksissa kun tuloksia tarkastellaan kymmenkantaisella logaritmilla. Myöskään solmut jotka ovat nieluja eivät erotu jakaumasta lainkaan arvostuksen pudotuksena.

Modifioidun HITS-algoritmin auktoriteettiarvojen visualisointi on esitetty kuvassa 5.7. Tässä solmun koko ja väri kuvaavat sen auktoriteettiarvoa, ja mitä korkeampi on kyseisen solmun arvostus, sitä suurempi ja väriltään lähempänä mustaa se on graafissa.

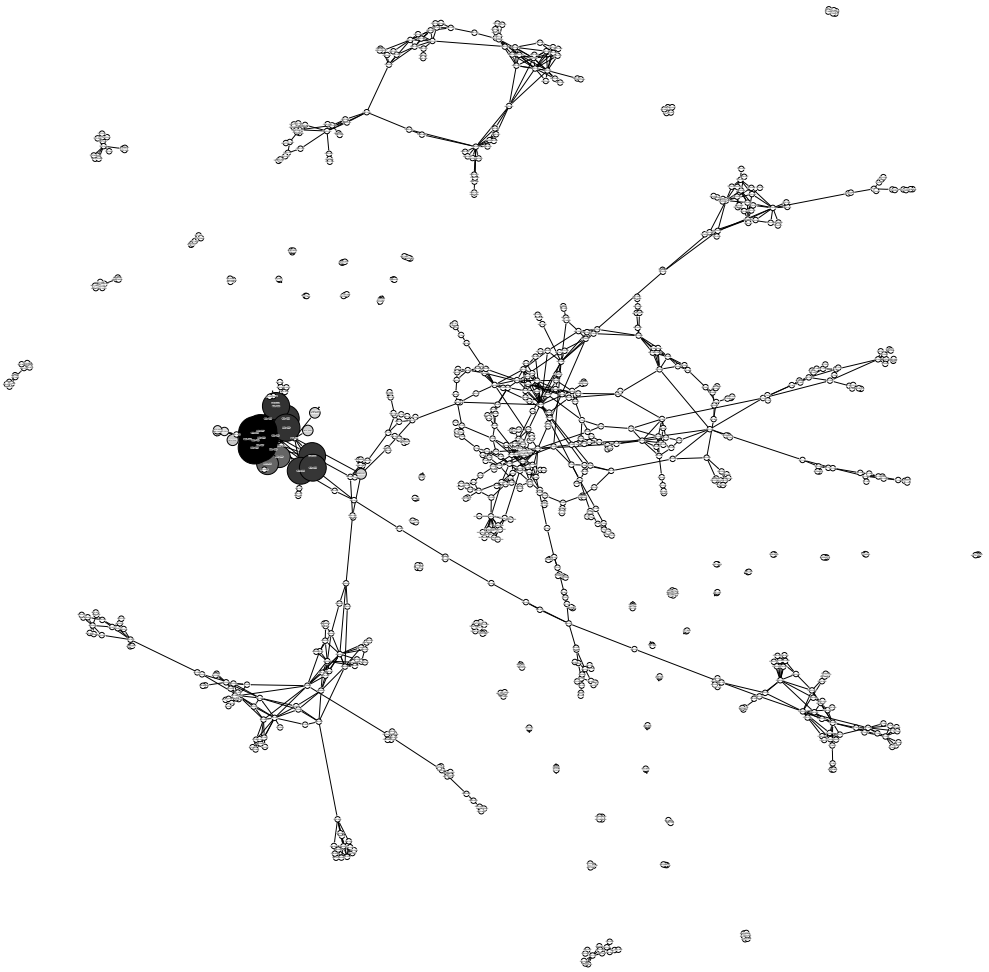
Kuvasta nähdään, että arvostus kasautuu täysin yhteen osaan graafia. Koska auktoriteetti- ja hubiarvot kytkeytyvät toisiinsa, on tarpeen ottaa myös hubiarvojen ominaisuudet tarkasteltaessa auktoriteettiarvoja.

Kurssien hubiarvot parametrilla, $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$ ovat esitettynä taulukossa 5.7. Tulokset on järjestetty parametrilla $\xi = 0,99$ saatavien tulosten mukaiseen suuruusjärjestykseen.

Taulukko 5.7: Kurssien 15 suurinta hubiarvoa modifioidulla HITS-algoritmilla parametreilla $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$

Hubi (log_{10})					
Kurssi	$\xi = 0,85$	$\xi = 0,95$	$\xi = 0,99$	Esitietoja	Esitietona
MOL-1330	-0,71	-0,71	-0,71	0	15
MOL-1310	-0,72	-0,72	-0,72	0	14
MOL-1320	-0,72	-0,72	-0,72	0	14
MOL-1210	-0,74	-0,74	-0,74	0	15
MOL-1410	-0,98	-0,98	-0,98	0	10
MOL-1420	-1,00	-1,00	-1,00	0	7
MOL-1500	-1,88	-1,88	-1,88	4	2
MOL-6930	-1,89	-1,89	-1,89	0	1
MOL-1430	-2,51	-2,51	-2,51	0	1
MOL-3416	-2,75	-2,75	-2,75	1	1
MATP-1321	-4,98	-5,51	-6,22	1	12
MATP-1331	-4,99	-5,51	-6,23	3	11
SGN-1201	-5,02	-5,54	-6,26	0	17
MATP-1311	-5,03	-5,56	-6,28	0	12
BIO-1250	-5,09	-5,61	-6,33	1	11

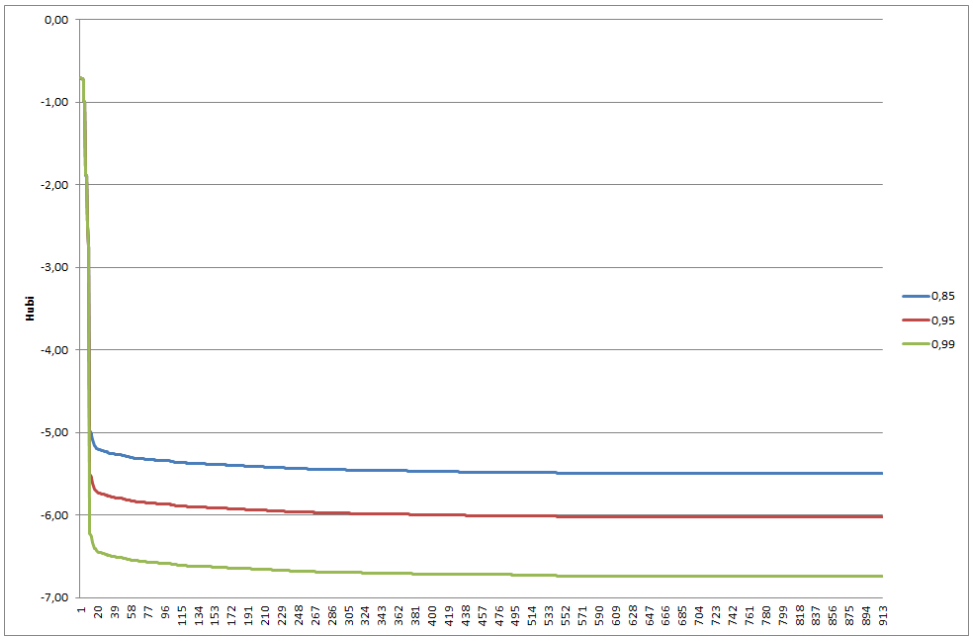
Hubiarvojen tulokset modifioidulla HITS-algoritmilla ovat täysin vastaavan kaltaiset kuin auktoriteettiarvotkin, joskin vielä entistäkin polarisoituneempia. Arvostetuimmat kurssit keräävät kukin lähes 20% kaikesta arvostuksesta, ja



Kuva 5.7: Kuva graafista, kun solmun koko ja väri kuvastaa kyseisen solmun auktoriteettiarvoa modifioidulla HITS-algoritmilla

kymmenen arvostetuimman kurssin yhteenlaskettu arvostus on peräti 99, 7%. Kaikki loput saavat hyvin pieniä arvoja, parametrin $\xi = 0,99$ tapauksessa alle 10^{-6}. Vaikka korkeimpia hubiarvoja saavista kursseista useimmat ovat todella monien kurssien esitietoina, on mukana peräti kolme kurssia jotka ovat ainoastaan yhden kurssin esitietona. Hubiarvoa keräävät kurssit ovat saman laitoksen kursseja kuin kurssit joille auktoriteettiarvot kasaantuivat, mikä viimeistään kielii ongelmista modifioidun HITS-algoritmin soveltamisessa. Auktoriteetti- ja hubiarvojen keskinäinen kytkentä näyttäisi kasaavan molempia arvostuksia tiettyyn osaan graafia.

Hubiarvojen jakauma modifioidulla HITS-algoritmilla on esitetty kuvassa 5.8.

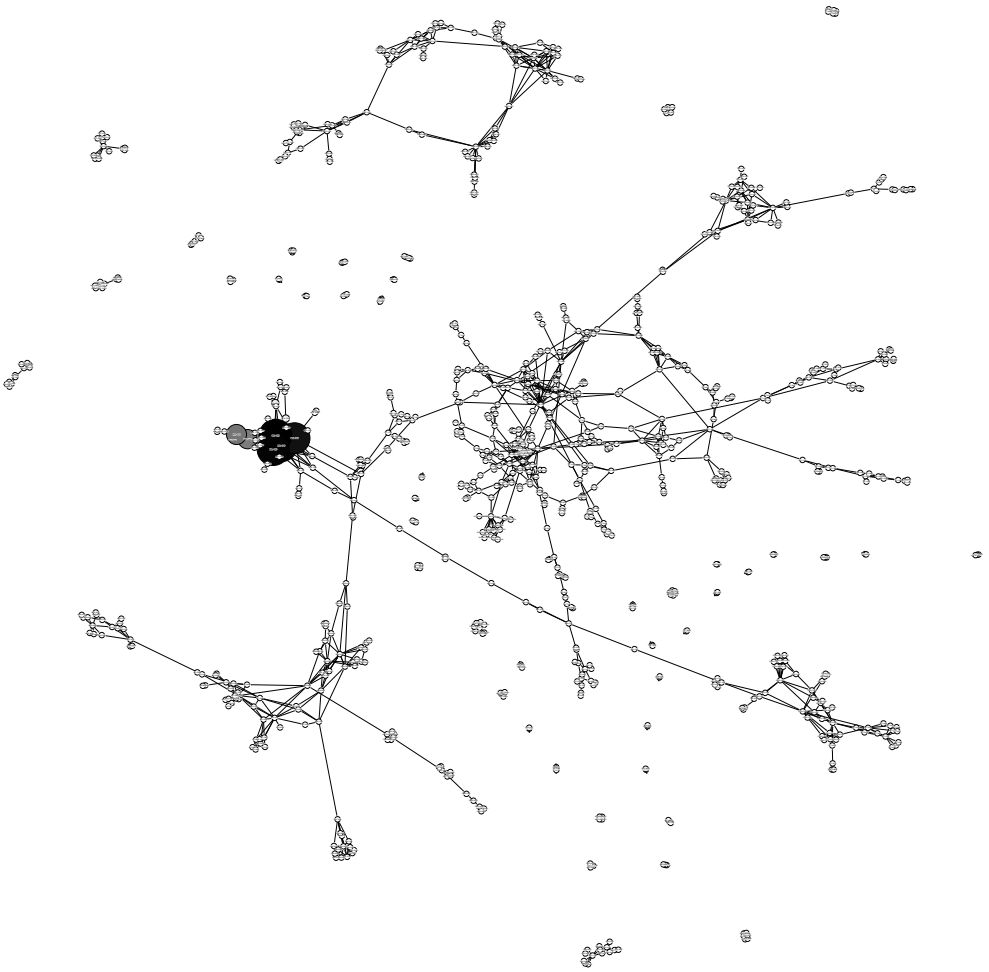


Kuva 5.8: Hubiarvojen jakauma kullakin parametrin ξ arvolla modifioidulla HITS-algoritmilla

Jakauma on käytännössä identtinen auktoriteettiarvojen jakauman kanssa, ja muutaman arvostetun kurssin jälkeen kaikkien muiden arvostus putoaa hyvin lähelle nollaa.

Myöskin kuvassa 5.9 esitetty visualisointi näyttää identtisellä auktoriteettiarvojen visualisoinnin kanssa, ja näyttää arvostuksen kasaantumisen samaan osaan graafia.

Nämä tulokset kertovat varsin yksiselitteisesti että modifoidun HITS-algoritmin käyttö tämän verkoston analysoinnissa antaa varsin ei-toivottavia tuloksia. Arvostus kasaantuu yhteen osaan graafia, joka sisältää hyvin tiheästi toisiinsa liittyneitä solmuja. Lähtödatan tarkastelu paljastaa, että Materiaaliopin laitoksen kurssien osalta esitietoketjut ovat melko täydellisiä, ja esimerkiksi materiaaliopin kolme peruskurssia ovat asetettu esitiedoiksi lähes kaikille kyseisen laitoksen kursseille. Näitä seuraa useita kursseja jotka toimivat taas monille kursseille



Kuva 5.9: Kuva graafista, kun solmun koko ja väri kuvastaa kyseisen solmun hubiarvoa modifioidulla HITS-algoritmilla

esitietona, joten näiden osalta muodostuvasta graafista tulee huomattavasti tiehämpi kuin koko esitietoketjuista keskimäärin.

Tämän ongelman käsittely herättää melko paljon kysymyksiä, ja asettaa melko perustavan laatuisia ongelmia algoritmin soveltamiselle. Ensinnäkin ongelmakohdassa käsiteltävä data oli kaikkein täydellisintä, joten ensimmäinen ajatus on että ongelma ei sinällään johdu datan rakenteesta, vaan pikemminkin sen epähomogeenisuudesta. Tämä aiheuttaa myös aika ison ongelman alkuperäistä käyttötarkoitusta eli internetin hakukoneita ajatellen, koska näiden tulosten valossa arvostukset olisivat hyvin helposti manipuloitavissa linkkispammin avulla. Tämä taas on arvostusalgoritmille erittäin huono ominaisuus, koska se asettaa kyseenalaiseksi sen objektiivisuuden. Valitettavasti tämän työn puitteissa ei ole mahdollista testailla modifioidun HITS-algoritmin toimintaa jollain homogeenisemmalla datalla, vaan tyydymme tutkimaan josko algoritmin toisella variaatiolla olisi mahdollista saada aikaan parempia tuloksia.

5.5 Satunnaistettu HITS

Kurssien auktoriteettiarvot parametrilla, $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$ ovat esitettynä taulukossa 5.8. Tulokset on järjestetty parametrilla $\xi = 0,99$ saatavien tulosten mukaiseen suuruusjärjestykseen.

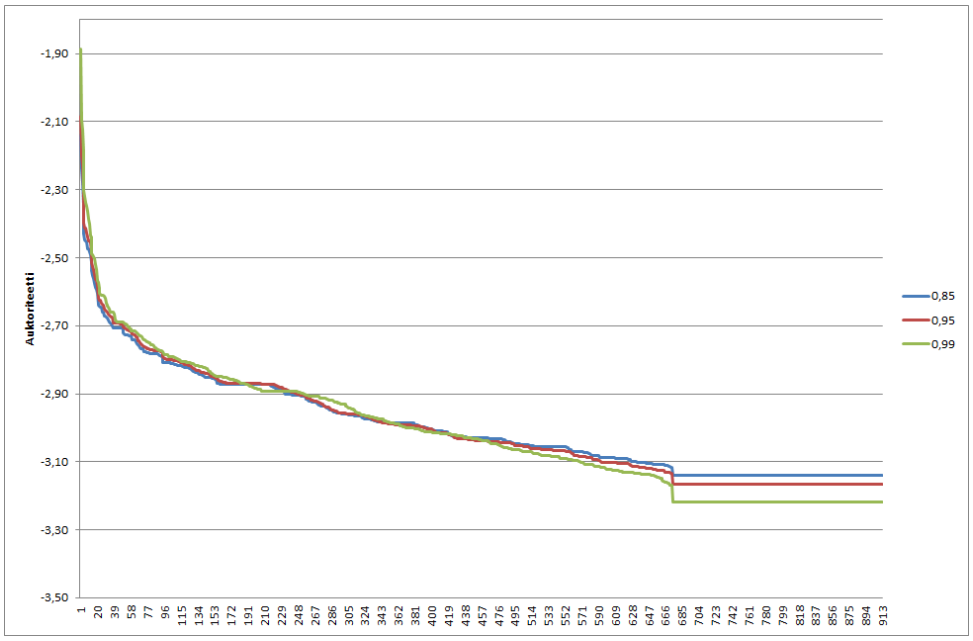
Taulukko 5.8: Kurssien 15 suurinta auktoriteettiarvoa satunnaistetulla HITS-algoritmilla parametreilla $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$

Auktoriteetti (log_{10})					
Kurssi	$\xi = 0,85$	$\xi = 0,95$	$\xi = 0,99$	Esitietoja	Esitietona
BIO-4700	-2,14	-2,09	-1,89	16	0
OHJP-3600	-2,28	-2,24	-2,10	11	0
PAK-2070	-2,24	-2,20	-2,13	13	0
SMG-4550	-2,35	-2,31	-2,19	6	0
TETA-4026	-2,45	-2,41	-2,30	11	0
MAT-33351	-2,45	-2,41	-2,32	7	0
RTEK-3880	-2,43	-2,40	-2,34	12	0
TETA-4020	-2,49	-2,46	-2,35	10	0
ELEP-4510	-2,47	-2,44	-2,36	9	3
OHJ-1500	-2,48	-2,45	-2,39	9	0
KEM-4050	-2,45	-2,43	-2,40	6	0
BME-4606	-2,47	-2,45	-2,44	5	0
ENER-3010	-2,51	-2,49	-2,44	8	1
RTEK-3630	-2,55	-2,53	-2,49	4	0
ACI-42086	-2,56	-2,53	-2,49	4	0

Tuloksia tarkasteltaessa huomataan välittömästi, ettei modifioidun

HITS-algoritmin kaltaista arvostuksen kasautumista tapahdu. Arvostetuimmat kurssit saavat hyvin saman suuruisia arvostuksia kuin Pagerankin tapauksessa, ja arvostetuimpien kurssien joukossa on useita samoja kursseja molemmilla algoritmeilla. Kärkeen nousevat syventävät, paljon esitietoja sisältävät kurssit. Vaikka listalle mahtuu myös kursseja joilla on vähemmän esitietoja, painottuvat tulokset Pagerankia vahvemmin kursseihin joilla on paljon esitietoja. Vastaavasti kuten Pagerankin tapauksessa, suuremmilla parametrin ξ arvoilla esitietojen laatu nousee määrää merkittävämmäksi tekijäksi, ja vähemmän esitietoja omaavat kurssit nousevat listalla.

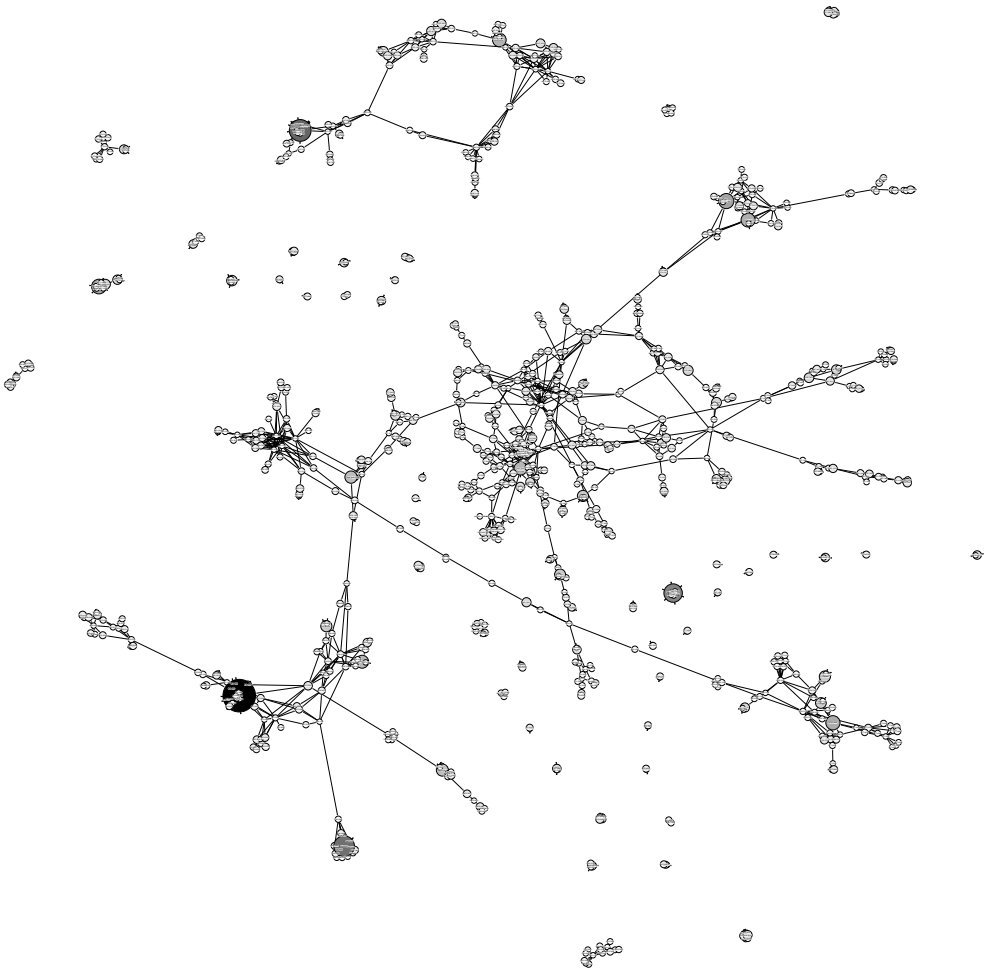
Auktoriteettiarvojen jakauma satunnaistetulla HITS-algoritmilla on esitetty kuvassa 5.10.



Kuva 5.10: Auktoriteettiarvojen jakauma kullakin parametrin ξ arvolla satunnaistetulla HITS-algoritmilla

Jakauma muistuttaa hieman eksponenttijakaumaa, mutta arvostetuimpien kurssien jälkeen laskee melko jyrkästi, jonka jälkeen laskee tasaisesti. Kurssit jotka ovat lähteitä saavat kaikki saman arvon, ja mielenkiintoisena ilmiönä tämä vakioarvo on selkeästi matalampi kuin seuraavaksi pienin arvostus. Tämän pudotuksen suuruus myös kasvaa parametrin ξ mukana, ja parametrilla $\xi = 0,99$ erotus lähteiden ja seuraavaksi vähiten arvostettujen kurssien välillä on jo varsin merkittävä.

Satunnaistetun HITS-algoritmin auktoriteettiarvojen visualisointi on esitetty kuvassa 5.11. Tässä solmun koko ja väri kuvaavat sen auktoriteettiarvoa, ja mitä korkeampi on kyseisen solmun arvostus, sitä suurempi ja väriltään lähempänä mustaa se on graafissa.



Kuva 5.11: Kuva graafista, kun solmun koko ja väri kuvastaa kyseisen solmun auktoriteettiarvoa satunnaistetulla HITS-algoritmilla

Kuvasta nähdään, että korkeimmat arvostukset menevät samoihin paikkoihin kuin Pagerankillakin, mutta koska pudotus näiden jälkeen on jyrkempi, ei keskitason arvostukset juurikaan erotu kuvasta.

Kurssien hubiarvot parametrilla, $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$ ovat esitettynä taulukossa 5.9. Tulokset on järjestetty parametrilla $\xi = 0,99$ saatavien tulosten mukaiseen suuruusjärjestykseen.

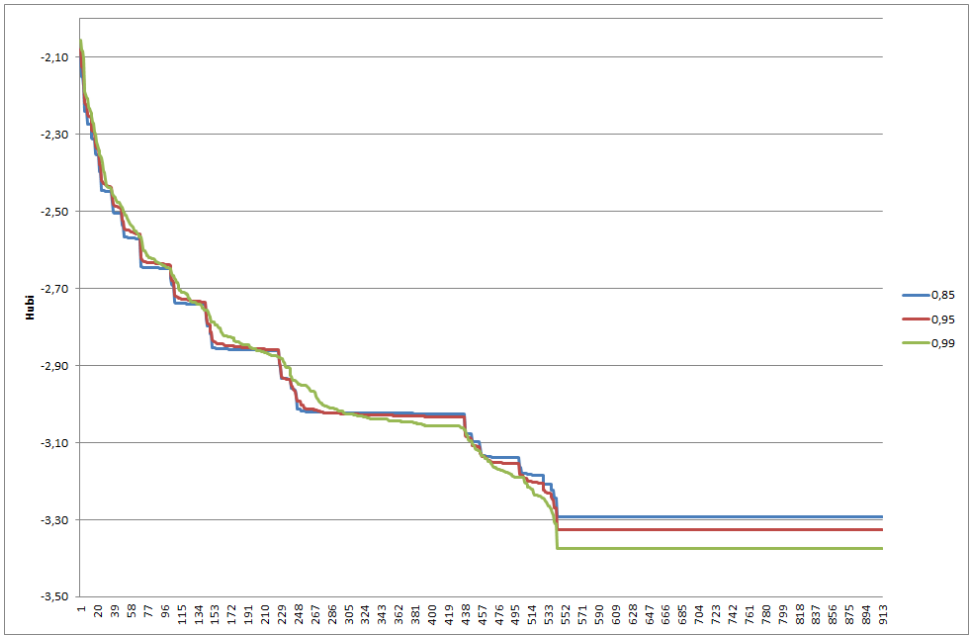
Taulukko 5.9: Kurssien 15 suurinta hubiarvoa satunnaistetulla HITS-algoritmilla parametreilla $\xi = 0,85$, $\xi = 0,95$ ja $\xi = 0,99$

Hubi (log_{10})					
Kurssi	$\xi = 0,85$	$\xi = 0,95$	$\xi = 0,99$	Esitietoja	Esitietona
SGN-1201	-2,10	-2,08	-2,06	0	17
MOL-1330	-2,15	-2,12	-2,08	0	15
MOL-1210	-2,15	-2,13	-2,08	0	15
MOL-1310	-2,18	-2,15	-2,10	0	14
MOL-1320	-2,18	-2,15	-2,10	0	14
MATP-1321	-2,24	-2,22	-2,19	1	12
BIO-1250	-2,27	-2,25	-2,20	1	11
TETA-1010	-2,24	-2,22	-2,20	0	12
MATP-1311	-2,24	-2,22	-2,21	0	12
OHJ-1150	-2,27	-2,25	-2,23	1	11
OHJ-1100	-2,27	-2,25	-2,23	1	11
MAT-20501	-2,27	-2,26	-2,24	0	11
RTT-1020	-2,28	-2,26	-2,25	0	11
MOL-1410	-2,31	-2,29	-2,25	0	10
SGN-1157	-2,31	-2,29	-2,27	0	10

Vastaavasti kuten auktoriteettiarvoilla, ei hubiarvoillekaan tapahdu vastaavaa kasautumista satunnaistetulla HITS-algoritmilla kuten modifioidulla HITS-algoritmilla. Kärkeen nousevat peruskurssit, jotka toimivat useille kursseille esitietona, ja näistä kaikilla on vähintään 10 esitietoa. Osa näistä kursseista on samoja kuin Cheirankin tapauksessa, mutta tässä painottuu selkeästi vahvemmin kurssit jotka toimivat useille kursseille esitietona. Tämä on seurausta näiden algoritmien eroista perusajatuksen tasolla: Cheirankin mukaan arvokkaimpia hubeja ovat ne jotka osoittavat hyviin hubeihin, ja HITS-algoritmin tapauksessa arvokkaimmat hubit ovat ne jotka osoittavat hyviin auktoriteetteihin. Näin ollen on sovelluskohteesta kiinni kumpi menetelmä antaa paremmin käsiteltävää ilmiötä kuvaavia tuloksia.

Hubiarvojen jakauma satunnaistetulla HITS-algoritmilla on esitetty kuvassa 5.12.

Satunnaistetulla HITS-algoritmilla saatujen hubiarvojen jakauma on varsin erilainen muiden algoritmien tuottamiin jakaumiin verrattuna. Ensimmäiseksi



Kuva 5.12: Hubiarvojen jakauma kullakin parametrin ξ arvolla satunnaistetulla HITS-algoritmilla

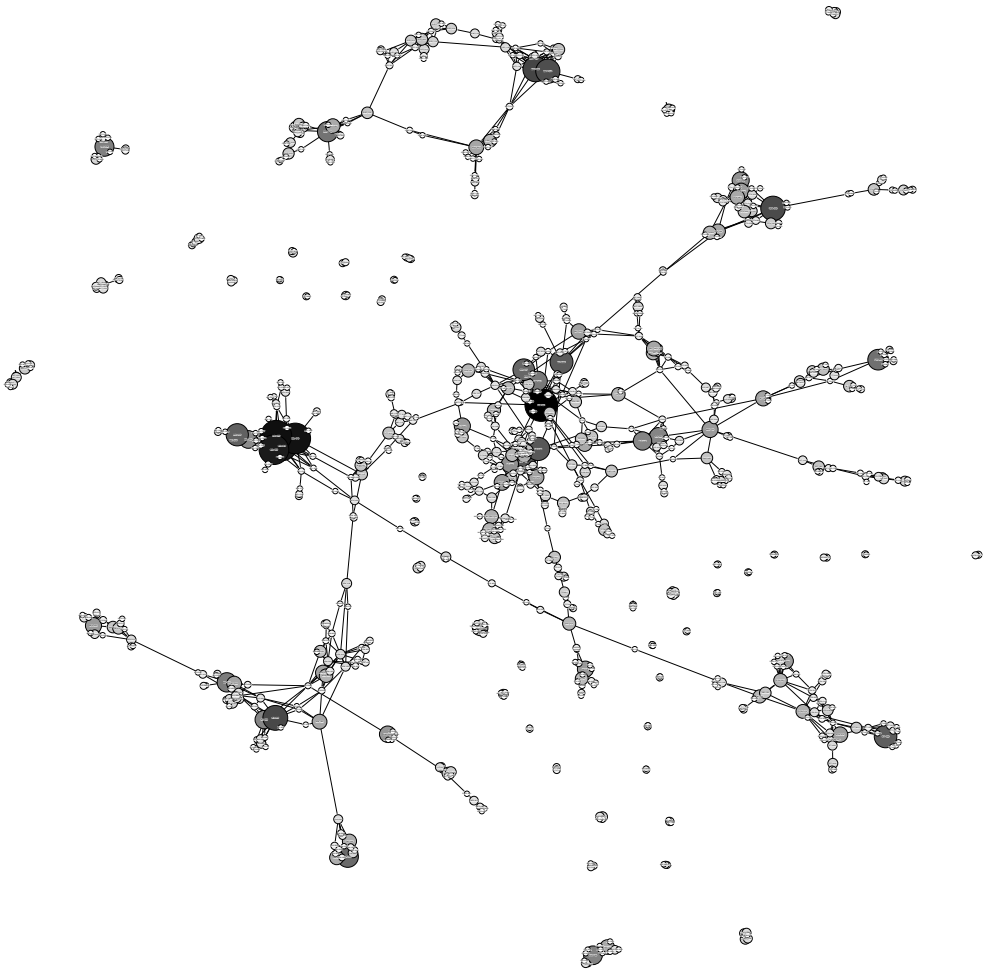
silmään pistää parametrin ξ vaikutus: pienemmillä parametrin ξ arvoilla jakaumassa on selkeitä portaita, mutta arvolla $\xi = 0,99$ nämä portaat häviävät ja jakauma on melko sileä. Toisekseen jakauma eroaa kaikkein selkeiten eksponenttijakaumasta. Alkupäässä hubiarvojen jakauma muistuttaa jokseenkin eksponenttijakaumaa, mutta puolessa välissä tapahtuu melko jyrkkä pudotus, jota seuraa huomattava pudotus nieluja kuvaavaan vakioarvoon.

Satunnaistetun HITS-algoritmin hubiarvojen visualisointi on esitetty kuvassa 5.13. Tässä solmun koko ja väri kuvaavat sen hubiarvoa, ja mitä korkeampi on kyseisen solmun arvostus, sitä suurempi ja väriltään lähempänä mustaa se on graafissa.

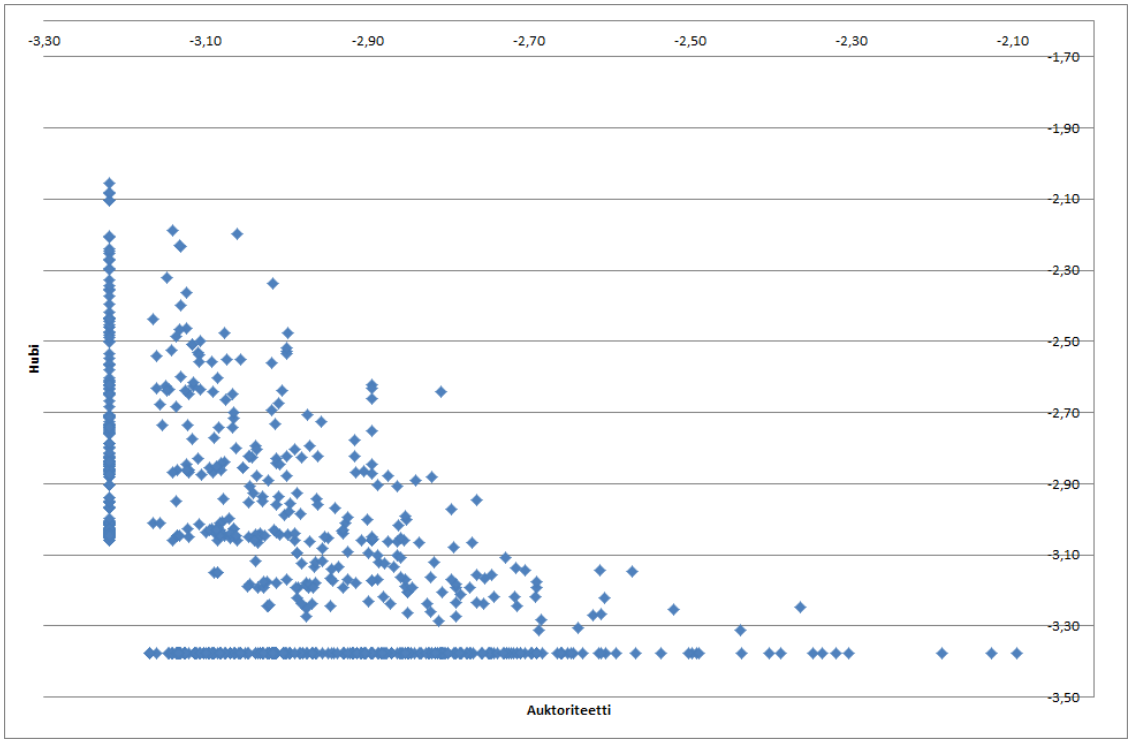
Kuvasta nähdään että korkeita hubiarvoja esiintyy paljon alueella mihin modifioidun HITS-algoritmin tapauksessa koko arvostus kasaantui. Kuitenkin arvostusta löytyy myös muista osista graafia, ja korkealle arvostettuja solmuja löytyy hyvin useasta osasta graafia.

Auktoriteetti- ja hubiarvojen yhteisjakauma satunnaistetulla HITS-algoritmilla on esitetty kuvassa 5.14. Koska tarkasteltavan ilmiön luonteen vuoksi on loogista tarkastella enemmän linkkirakenteen mukaista etenemistä verkostossa, on tässäkin tapauksessa tarkasteltavaksi skenaarioksi valittu auktoriteetti- ja hubi arvot jotka on saatu parametrilla $\xi = 0,99$.

Auktoriteetti- ja hubiarvojen yhteisjakaumasta nähdään selkeästi kummallakin akselilla minimiarvostuksen eli nielujen ja lähteiden selkeä ero toiseksi matalimpaan arvostukseen. Vastaavasti vasempaan alanurkkaan lähelle molempien



Kuva 5.13: Kuva graafista, kun solmun koko ja väri kuvastaa kyseisen solmun hubiarvoa satunnaistetulla HITS-algoritmilla



Kuva 5.14: Kurssien auktoriteeti- ja hubiarvojen yhteisjakauma satunnaistetulla HITS-algoritmilla parametrilla $\alpha = 0,99$

minimiä jää melko suuri aukko, johon ei osu yhtäkään kurssia. Muutoin jakauma on hyvin samankaltainen kuin Pagerankin ja Cheirankin yhteisjakauma, ja auktoriteetti- ja hubiarvojen välinen riippuvuus vaikuttaisi myös olevan jokseenkin käänteinen. Hajonta on kuitenkin huomattavasti pienempää kuin Pagerankin ja Cheirankin tapauksessa, ja suurin osa kursseista sijaitsee jakaumassa varsin tiiviillä alueella, joka muodostaa jakaumaan kolmiomaisen alueen.

Edellä esitetyt tulokset indikoivat selvästi, että HITS-algoritmin modifioitu variaatio toimii tämän datan tapauksessa varsin epäjohdonmukaisesti. Algoritmin satunnaistettu variaatio sen sijaan antaa järjellisempiä tuloksia, ja antaa merkityksellistä tietoa verkoston rakenteesta. Tämä ilmiö on täysin sama mikä havaittiin aiemmin pienempää esimerkkigraafia tarkasteltaessa. Näin ollen voidaan melko luottavaisin mielin todeta, että HITS-algoritmia sovellettaessa on suotavaa käyttää algoritmin eksponentiaalista- tai satunnaistettua variaatiota. Nämä molemmat antavat melko samansuuntaisia tuloksia, mutta eroja on kuitenkin sen verran että on syytä tarkastella kussakin sovelluskohteessa kumpi näistä on soveltuvampi. Yleisesti algoritmin satunnaistettu versio on kuitenkin lopulta suositeltavin versio HITS-algoritmista, sillä tämän edut laskentanopeudessa verrattuna eksponentiaaliseen variaatioon ovat merkittävät. Tämä ei myöskään aseta vaatimuksia tarkasteltavalle graafille, joten graafin ei tarvitse olla vähintään heikosti yhtenäinen kuten eksponentiaalisen HITS-algoritmin tapauksessa.

5.6 Johtopäätöksiä

Edellisissä osioissa kävimme läpi tuloksia eri algoritmeilla esimerkkidatalle, joka koostui Tampereen teknillisen yliopiston vuoden 2010 opinto-oppaan esitietoketjuista, jotka muodostivat tarkasteltavan graafin. Nuolien suunta graafia määriteltäessä oli vapaavalintainen, ja tässä tapauksessa päädyttiin määrittelemään graafin nuolet niin, että esitietona olevasta kurssista lähtee nuoli kurssiin, jonka esitieto tämä on. Tämä suunta olisi voitu valita myös toisin, mutta tämän vaikutus eri arvostusalgoritmeihin olisi ollut lähinnä triviaali. Kun Pagerankia ja Cheirankia tarkastellaan yhdessä, on täysin irrelevanttia miten päin nuolet graafissa määritellään, koska suunnan muuttuessa nämä kaksi vaihtuvat toisiinsa. HITS-algoritmin tapauksessa nuolien suunnan määrityksellä on merkitystä, mutta tässä tapauksessa valittu suunta antaa tuloksia jotka ovat loogisia auktoriteetti- ja hubikäsitteiden kanssa. Vastaavalla käsitteillä Pagerankin voidaan ajatella kuvaavan auktoriteettia, ja Cheirankin hubia.

Modifioidun HITS-algoritmin epäjohdonmukaisia tuloksia lukuunottamatta muut algoritmit olivat melko yksimielisiä verkoston toimijoiden arvostuksesta. Korkeimmat auktoriteetti- ja Pagerank-arvot saivat syventävät kurssit, joilla on paljon esitietoja ja jotka ovat pitkien esitietoketjujen loppupisteitä. Vastaavasti Hubi- ja Cheirank arvoiltaan korkeimmaksi nousivat kurssit, jotka ovat peruskursseja ja toimivat useiden kurssien esitietoina, ja esitietoketjujen alkupisteinä. Auktoriteetti- ja Pagerank-arvoiltaan arvokkaimmat kurssit olivat hyvin pitkälti samoja, ja näiden osalta algoritmien välillä oli lähinnä nyanssieroja. Hubi- ja Cheirank-arvostuksissa sen sijaan oli enemmän eroja, ja nämä taas johtuivat algoritmien eri ajatusmalleista: HITS-algoritmilla hyvät hubit osoittavat hyviin auktoriteetteihin ja Cheirankilla hyvät hubit osoittavat hyviin hubeihin. Tämä aiheutti hieman eroja kurssien arvojärjestykseen, ja näin ollen kussakin sovelluskohteessa on syytä miettiä kumpi näistä antaa relevantimpaa tietoa kyseiseen ilmiöön liittyen.

Pohdittaessa algoritmien hyödyntämistä esimerkkidatan tapauksessa, nousee ensimmäiseksi kysymykseksi kumpi algoritmeista, Pagerank (jatkettuna Cheirankilla) vai HITS kuvaa paremmin tarkasteltavaa ilmiötä. Nykyisen kaltaisessa internetissä hubisivustojen merkitys on laskenut, koska relevanttia tietoa on entistä helpommin löydettävissä ja saatavilla. Tämä, ja toisaalta Googlen vahva asema hakukonemarkkinoilla on johtanut Pagerankin "voittoon"tällä markkinalla, eikä HITS-algoritmia ole tällä hetkellä käytössä oikeastaan missään merkittävässä hakukoneessa. Kurssien esitietoketjuja tarkasteltaessa on kuitenkin hyödyllistä tarkastella arvostusta kahdessa ulottuvuudessa, sillä näin muodostuvassa graafissa ketjujen alku- ja loppupään kurssit eroavat selkeästi

toisistaan. Ajatuksen tasolla HITS-algoritmin ajatus hubeista jotka osoittavat hyviin auktoriteetteihin tuntuu tässä tapauksessa hieman luontevammalta, mutta toisaalta tulosten valossa on hyvin vaiketa sanoa voidaanko algoritmeista sanoa kumpi antaa kuvaavampaa informaatiota verkoston toimijoista.

Tulosten aikaansaannin ja tulkinnan ohella on syytä miettiä myös miten näitä tuloksia voisi hyödyntää, ja minkälaisia johtopäätöksiä niiden perusteella voitaisiin tehdä. Kuten aiemmin todettiin, sinällään yksittäisen opiskelijan liikkumiseen kurssilta toiselle on näiden tulosten pohjalta vaikea ottaa kantaa, mutta joitain valistuneita ajatuksia on mahdollista esittää. Periaatteessa alottavan opiskelijan olisi syytä etsiytyä kursseille jotka ovat korkealla hubi- ja Cheirank-arvoissa, ja jatkaa näistä esitietoketjujen mukaisesti kasvavan auktoriteetti- ja Pagerank-arvon suuntaan. Kun opiskelija saapuu ketjun päähän, tulisi hänen siirtyä taas johonkin korkean hubi- ja Cheirank-arvon omaavaan kurssiin. Tämä kuitenkaan ei ota huomioon opiskelijan omia mieltymyksiä tai opintokokonaisuuksien vaatimuksia, mutta toisaalta nämä vaatimukset ovat osatekijä joka määrittää esitietoketjut ja sitä kautta kunkin kurssin saaman arvostuksen.

Toinen mahdollinen sovelluskohde on kurssitarjonnan muokkaaminen. Periaatteessa kurssitarjontaa uudistaessa tulisi suosia kursseja jotka ovat jommassa kummassa arvostuksessa korkealla, ja varsinkin korkean hubi- tai Cheirank-arvon omaamien kurssien poistamisella kurssitarjonnasta saattaisi olla vaikutusta myöhempiin kursseihin. Toisaalta kurssien tarjonta ja sisältö ei ole staattista, ja eri asioiden sisällyttäminen eri kursseille ja tätä kautta esitietoketjujen muovautuminen tietenkin muuttaa kurssien arvostusta. Myöskään kurssien todellista arvoa ei voida näin suoraviivaisesti määrittää, sillä näihin vaikuttavat joka tapauksessa kyseistä kurssia tarjoavan laitoksen muutkin tavoitteet, kuten om tutkimus.

Kolmas mahdollinen sovelluskohde näille arvostuksille on kurssien arvosanojen painotus. Auktoriteetti- ja Pagerank-arvoissa korkealle nousevat kurssit ovat kursseja, jotka vaativat lukuisien (ja usein myös vaativien) esitietokurssien suorittamista, jolloin voidaan perustellusti sanoa että näiden kurssien painoarvo opiskelijan arvostelussa tulisi olla korkeampi. Nykyinen opintopistemäärään perustuva painotus on kohtalaisen relevantti työmäärän mittari, mutta toisaalta työmäärä ei yksin ole kovinkaan hyvä mittari kurssien todelliselle vaativuudelle. Tämä edellyttäisi täydellisiä esitietoketjuja, ja myöskin objektiivista tarkastelua todellisista esitietovaatimuksista manipuloinnin estämiseksi. Ajatuksen tasolla tämä on kuitenkin varsin mielenkiintoinen vaihtoehto perinteiselle työmäärään perustuvalle vaativuusmittaukselle.

Tässä työssä tarkasteltiin yksinkertaista, yksimoodista graafia, jonka kaikki nuolet olivat samanpainoisia. Useissa sovelluskohteissa nämä kaikki oletukset eivät

välttämättä ole voimassa, ja tarkasteltavat graafit voivat olla hyvinkin monenlaisia. Tässä käsiteltyjen arvostusalgoritmien hienous piilee kuitenkin siinä, että ne eivät sinällään aseta mitään vaatimuksia näille ominaisuuksille, vaan nämä sattuivat olemaan tarkasteltavan esimerkkigraafin ominaisuuksia. Kaikki käsitellyt konseptit yleistyvät suoraan niin moninkertaisille, useampimoodisille kuin painotetuillekin graafeille. Näissä tapauksissa pohjimmiltaan ainoastaan vieruspistematriisi muuttuu, ja mikään esitetyistä teorioista tai algoritmeista ei edellytä että vieruspistematriisi olisi binäärinen. Näin ollen nämä kaikki algoritmit ovat käytettävissä sellaisenaan myös monimutkaisemmille graafeille.

6. YHTEENVETO

Tässä työssä on esitetty eri arvostusalgoritmeja verkoston toimijoiden asettamiseksi tärkeysjärjestykseen. Näistä algoritmeista keskityttiin kahteen keskeisimpään: Pagerankiin sekä HITS-algoritmiin eri variaatioineen. Nämä kaksi ovat ensimmäisiä ja merkittävimpiä verkoston linkkirakenteeseen perustuvia arvostusalgoritmeja, joiden konsepteihin useimmat myöhemmin kehitetyistä algoritmeista perustuvat. Nämä molemmat ovat alunperin kehitetty internetin hakukoneiden tulosten laittamiseksi järjestykseen, mutta näitä voidaan soveltaa helposti käytännössä minkälaiselle verkostolle tahansa.

Verkostojen kuvaamiseen käytetään graafeja, ja graafiteoria tarjoaakin hyvät työkalut verkostojen visualisointiin ja perusominaisuuksien tarkasteluun. Graafiteoria kytkeytyy kiinteästi matriiseihin, ja arvostusalgoritmit pohjautuvatkin hyvin pitkälti graafien matriisien käsittelyyn. Yhdessä graafiteoria ja matriisiteoria antavat arvostusalgoritmien käsittelylle elegantin ympäristön, jossa molempien ominaisuudet nivoutuvat toisiinsa muodostaen selkeän kokonaisuuden. Teoreettinen pohja on kuitenkin vain puolet (tai pikemminkin murto-osa) itse analyysista, ja oikeiden menetelmien käyttö, sekä erityisesti tulosten tulkinta ovat vaiheita jotka asettavat todelliset haasteet analyysiprosessissa. Jotta analyysissa voidaan käyttää soveltuvimpia menetelmiä, sekä esittää mahdollisimman laadukkaita tulkintoja, edellytetään analysoijalta vahvan teoriapohjan lisäksi hyvää käsitystä tarkasteltavasta ilmiöstä, ja ymmärrystä miten eri menetelmät käyttäytyvät kyseisen ilmiön analysoinnissa.

Esitettyjä arvostusalgoritmeja on testattu käytännössä tutkimalla Tampereen teknillisen yliopiston vuoden 2010 opinto-oppaan esitietoketjuja. Nämä esitietoketjut muodostavat graafin, jossa kurssit ovat solmuja ja kurssien väliset esitietoriippuvuudet solmujen välisiä nuolia. Näin muodostuvaa graafia käsiteltiin eri arvostusalgoritmeilla, ja tutkittiin graafiteorian tunnuslukuja ja erilaisia visualisointeja saadaksemme mahdollisimman hyvän käsityksen verkoston rakenteesta, sekä eri toimijoiden merkityksestä siinä. Eri algoritmit toimivat tämän verkoston tarkastelussa pääsääntöisesti hyvin, ja tämä osoitti algoritmien käyttökelpoisuutta erilaisten ilmiöiden tutkimisessa, sillä vaikka kurssien esitiedot ja internetin linkkirakenne ovat ilmiönä melko erilaisia, näitä voidaan mallintaa erittäin hyvin melko vastaavilla graafeilla.

Saadut tulokset ja niiden tulkinta olivat melko suoraviivaisia, mutta tulosten hyödyntäminen herätti monenlaisia pohdintoja. Pohjimmiltaan tässä työssä on enimmäkseen kyse akateemisesta mielenkiinnosta ja kokeilunhalusta, ja täysin suoran hyödyn saaminen näistä tuloksista ei ole ongelmatonta. Kuitenkin näistä heräsi useampia ajatuksia miten tuloksia voisi hyödyntää, ja minkälaisia haasteita tulosten hyödyntäminen missäkin tapauksessa asettaisi.

KIRJALLISUUTTA

[1] Langville, A.N. & Meyer, C.D. (2006) *Google's PageRank and beyond: The science of seach engine rankings*. Princeton University Press

[2] Wasserman, S. & Faust, K. (1994) *Social network analysis: Methods and applications.* New York, Cambridge University Press

[3] Eldén, L. (2007) *Matrix methods in data mining and pattern recognition (Fundamentals of algorithms)*. Society for Industrial and Applied Mathematics

[4] Ruohonen, K. (2006) *Graafiteoria*. Tampere, Tampereen teknillisen yliopiston opetusmoniste no. 5, uusi sarja.

[5] Miilumäki, T. (2010) *Web-pohjaisten sosiaalisten verkostojen analyysimenetelmät*. Tampere, Tampereen teknillinen yliopisto, diplomityö.

[6] Meyer, C.D. (2000) *Matrix analysis and Applied Linear Algebra*, Philadelphia, SIAM

[7] Smith, R. (2007) *Matrix Analysis and Computation*. Santa Barbara, University of California, lecture notes

[8] Blondel, V. D. & Guillaume, J-L & Lambiotte, R. & Lefebvre, E. (2008) *Fast unfolding of communities in large networks*. J. Stat. Mech. P10008

[9] Jacomy, M. & Heymann, S. & Venturini, T. & Bastian, M. (2011) *ForceAtlas2, A Graph Layout Algorithm for Handy Network Visualization.* $http://webatlas.fr/tempshare/ForceAtlas2\_Paper.pdf$, viitattu 13.5.2012

[10] Brin, S. & Page, L. (1998) *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks and ISDN Systems 30: $107 - 117$

[11] Kleinberg, J. (1999) *Authoritative sources in a hyperlinked environment*, Journal of the ACM 46 (5): $604 - 632$.

[12] Ermann L. & Chepelianskii A.D. & Shepelyansky D.L. (2011) *Towards two-dimensional search engines*, J. Phys. A. Math. Theor. 45 (2012) 275101

[13] Lempel, R. & Mora, S. (2000) *The stochastic approach for link-structure analysis (SALSA) and the TKC effect*, The Ninth International World Wide Web Conference, New York, ACM Press

[14] Farahat, A. & Lofaro, T. & Miller, J.C. & Rae, G. & Ward, L.A. (2006) *Authority rankings from HITS, Pagerank and SALSA: existence, uniqueness, and effect of initialization.* SIAM Journal on Scientific Computing, 27(4):1181-201

[15] Ng, A.Y. & Zheng, A.X. & Jordan, M.I. (2001) *Stable algorithms for link analysis*, ACM

[16] Benzi, M. & Estrada, E. & Klymko, C. (2012) *Ranking hubs and authorities using matrix functions*, CoRR, Vol. abs/1201.3120

[17] Miller, J.C. & Rae, G. & Schaefer, F. (2001) *Modifications of Kleinberg's HITS Algorithm Using Matrix Exponentiation and Web Log Records*, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, p 444-445